

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**Факультет електроніки
Кафедра промислової електроніки**

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ ПРАКТИЧНИХ ЗАНЯТЬ
з дисципліни «Цифрові інформаційні системи»
для студентів спеціальності
171 Електроніка
спеціалізації Електронні системи**

**Київ
НТУУ «КПІ»
2016**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**Факультет електроніки
Кафедра промислової електроніки**

**ДО ВИКОНАННЯ ПРАКТИЧНИХ ЗАНЯТЬ
з дисципліни «Цифрові інформаційні системи»
для студентів спеціальності
171 Електроніка
спеціалізації Електронні системи**

*Рекомендовано Вченою радою
факультету електроніки НТУУ «КПІ»
протокол № 10/2016 від 31.10.2016
(протокол №, дата)*

Методичні вказівки до виконання практичних занять з дисципліни «Цифрові інформаційні системи» для студентів спеціальності 171 Електроніка, спеціалізації Електронні системи / Уклад.: Осипенко К.С. – К.: КПІ, 2016. – 118 с.

Укладач: Осипенко Катерина Сергіївна, канд. техн. наук.

Рецензент: Новський Володимир Олександрович, с.н.с., докт. техн. наук

Відповідальний редактор: Терещенко Тетяна Олександрівна, проф.,
докт. техн. наук

ЗМІСТ

| | |
|--|----|
| Вступ | 4 |
| 1. Практичне заняття № 1. Представлення чисел в різних системах числення | 5 |
| 2. Практичне заняття № 2. Виконання найпростіших арифметичних операцій | 10 |
| 3. Практичне заняття № 3 – 4. Алгебра логіки | 13 |
| 4. Практичне заняття № 5. Мінімізація логічних функцій методом Квайна | 26 |
| 5. Практичне заняття № 6. Факторизація | 34 |
| 6. Практичне заняття № 7. Перетворювачі кодів (дешифратори) | 37 |
| 7. Практичне заняття № 8. Мультиплексори – демультиплексори | 43 |
| 8. Практичне заняття № 9. Суматори – компаратори | 51 |
| 9. Практичне заняття № 10. Тригери | 59 |
| 10. Практичне заняття № 11 – 15. Скінченні автомати | 74 |
| 11. Практичне заняття № 16. Лічильники | 92 |

ВСТУП

Метою вивчення дисципліни «Цифрові інформаційні системи» є отримання студентами ґрунтовних знань з питань теорії та практики побудови цифрових інформаційних систем (ЦІС). Розглядається важливий клас ЦІС – автомати Мілі та Мура синхронного та асинхронного типів, синтез та аналіз їх структур, використання постійних запам'ятовуючих пристроїв (ПЗП), а також програмованих логічних інтегральних схем (ПЛІС) та побудова на їх основі цифрових пристроїв контролю, перетворення та відображення інформації.

Згідно з вимогами програми навчальної дисципліни студенти після засвоєння дисципліни мають продемонструвати такі результати навчання:

- знання з теорії представлення та мінімізації цифрових пристроїв, синтезу та аналізу цифрових автоматів за методикою Мілі та Мура синхронного та асинхронного типів із застосуванням сучасної інтегральної елементної бази;
- вміння використовувати теоретичні знання для розробки різноманітних структур ЦІС завадостійкого типу, проводити тестовий та функціональний контроль;
- навички застосувати набуті знання при виконанні курсових та випускної роботи, у практичній діяльності та наукових дослідженнях за фахом.

Поставлені цілі досягаються за рахунок широкого застосування в навчальному процесі обчислювальної техніки та сучасного програмного забезпечення, підвищення пізнавальної та творчої активності студентів.

Методичні вказівки містять інформацію по 16 практичних роботах. По кожній з робіт наведено послідовно тему, мету, завдання, теоретичні відомості, контрольні запитання та список рекомендованої літератури.

Практичне заняття № 1

Тема: Представлення чисел в різних системах числення.

Мета: Закріплення умінь і навичок переведення чисел між системами числення з різними основами.

Завдання

1. Записати десяткове число 105 у двійковій системі числення.
2. Перетворити числа 118, 57, 196, 238, що представлені у десятковій системі числення, в еквівалентні двійкові числа.
3. Перетворити числа десятикової системи числення 255 та 3959 в еквівалентні шістнадцяткові числа.
4. Перетворити числа десятикової системи числення 0,27 і 0,625 у двійкові з чотирма знаками після коми.
5. Перетворити числа $0,31_{10}$ та $0,761_{10}$ в шістнадцяткові з трьома знаками після коми.
6. Перевести десяткове число 25,179 у двійкове з трьома знаками після коми.
7. Записати десяткове число 69,42 в шістнадцятковій системі числення.
8. Перевести двійкове число 1011,01 в еквівалентне десяткове число.
9. Записати число $3F,9_{16}$ в десятковій системі числення.
10. Представити двійкове число 11110101,1011 у шістнадцятковій системі числення.
11. Записати число $C7,9H$ в двійковій системі числення.
12. Представити двійкове число 1101001 в зворотному коді.
13. Представити двійкове число 01101100 в додатковому коді.
14. Визначити додатковий код числа -28_{10} .

Теоретичні відомості

Система числення – це спосіб запису (зображення) чисел.

Системи числення, в яких ваговий коефіцієнт кожної цифри залежить від її положення у послідовності цифр, що зображає число, називаються *позиційними*. У *непозиційних* системах значення кожної цифри постійне і не залежить від місця її розташування в числі. Всі системи числення, які використовуються в цифровій схемотехніці, є позиційними.

При розгляді позиційних систем важливим виступає поняття *базису*. *Базис системи числення* – це послідовність чисел, яка задає значення (вагу) кожної цифри в залежності від місця її розміщення.

У загальному випадку для позиційних систем числення базис можна записати в вигляді послідовних членів геометричної прогресії:

$$P^{-m}, \dots, P^{-2}, P^{-1}, P^0, P^1, P^2, \dots, P^n, \dots,$$

де число P називається *основою системи числення*.

Сукупність різних цифр, які використовуються в позиційній системі числення для запису чисел, називається *алфавітом системи*.

Кількість цифр в алфавіті P -ічної системи числення дорівнює основі системи числення, починаючи з нуля. Тому алфавітом P -ічної системи числення є натуральний ряд чисел від нуля до P^{i-1} .

Переведення чисел з однієї позиційної системи числення в іншу. Якщо число A представлено в P -ічній системі числення, то його для переведення в десяткову систему числення його необхідно представити в формі

$$A = a_n P^n + a_{n-1} P^{n-1} + \dots + a_1 P^1 + a_0 P^0 + a_{-1} P^{-1} + a_{-2} P^{-2} + \dots + a_k P^{-k}.$$

Для отримання значення цього багаточлена, записаного в десятковій системі числення, необхідно число P і коефіцієнти при ступенях P (цифри алфавіту P -ічного числа) записати у вигляді десяткових чисел, і всі обчислення провести в десятковій системі.

Переведення цілого числа з десяткової системи числення в P -ічну. Для виконання подібних перетворень використовується декілька способів. Один з них полягає у наступному. Запишемо відоме число A_{10} , представлене в десятковій системі числення, в умовній P -ічній системі числення, де коефіцієнти a_n поки що невідомі:

$$A_{10} = a_n \cdot P^n + \dots + a_1 \cdot P^1 + a_0.$$

Розділивши праву і ліву частини на P , отримаємо ціле число

$$a_n \cdot P^{n-1} + \dots + a_1$$

і залишок, величина якого не перевищує значення $P - 1$. Таким шляхом отримується остання цифра запису числа в P -ічній системі числення. Не зовсім зрозуміло: залишок – це і є остання цифра. Виконуючи аналогічне ділення десяткового числа n разів, можемо отримати всі невідомі коефіцієнти P -ічної системи числення.

Для переведу дробової частини з десяткової системи числення в P -ічну необхідно представити її у розгорнутому вигляді:

$$A_{10} = a_{-1} \cdot P^{-1} + a_{-2} \cdot P^{-2} + \dots + a_{-k} \cdot P^{-k} + \dots$$

Перемножуючи ліву і праву частини виразу на P в правій частині виразу отримуємо:

$$a_{-1} + a_{-2} \cdot P^{-2} + \dots + a_{-k} \cdot P^{-k} + \dots$$

З отриманого результату можемо зробити висновок, що перша цифра a_{-1} дробової частини числа A в P -ічній системі числення дорівнює цілій частині результату перемноження десяткової дробової частини на P . Після чергового перемноження залишку дробової частини на P отримаємо значення a_{-2} :

$$(a_{-2} \cdot P^{-1} + \dots + a_{-k} \cdot P^{-k+1} + \dots) \cdot P.$$

Цей процес продовжується до тих пір, поки дробова частина результату перемноження лівої частини не стане рівною нулю або поки не буде виділений період повторності цифр.

Переведення числа з однієї недесяткової системи числення в аналогічну іншу виконується шляхом перетворення P -ічної системи числення в десяткову,

після чого виконується друга частина операції. Виключення складає лише переведення з двійкової системи числення в шістнадцяткову і навпаки.

Контрольні запитання

1. Дайте визначення системи числення. Наведіть приклади позиційних і непозиційних систем числення. Дайте пояснення.
2. Що Ви розумієте під терміном “алфавіт системи числення”? Наведіть приклади алфавітів двійкової системи числення; шістнадцяткової системи числення; системи числення з основою 20.
3. Скільки і яких цифр потрібно, щоб будь-яке число можна було записати у сімковій системі числення? В дванадцятковій?
4. Поясніть особливості використання схеми Горнера для переведення чисел з P -ічної системи числення в десяткову. Про схему Горнера нічого не було в теоретичних відомостях
5. Дайте пояснення особливостям переведення чисел з десятикової системи числення у P -ічну.
6. Дайте пояснення особливостям переведення дробових чисел з однієї системи числення в іншу. Наведіть конкретні приклади.

Рекомендована література

1. J. Eldon Whitesitt Boolean Algebra and Its Applications. Courier Corporation, 1995. 182 pp.
2. Алгебра і теорія чисел: Учб. посібник для студентів-заочників II курсу фіз.-мат. фак. пед. ін-тів (Н.А.Казачек та ін) / Під ред. Н.Я. Виленкина - 2-ге вид. М.: Просвіта, 1984. - 192 с.
3. Бендукідзе А.Д. Про системи числення// Квант - 1975 - № 8 – с. 59-61.

4. Берман Г.Н. Число і наука про нього. Загальнодоступні нариси з арифметики натуральних чисел. Вид. 3-є. М.: Фізматгіз, 1960. – 164 с.

5. Вайман А.А. Шумеро-вавілонська математика. III - I тисячоліття до н.е. М.: Вид. сх. літ., 1961. – 278 с.

6. Вигодський М.Я. Арифметика і алгебра в стародавньому світі. Вид. 2-є, випр. і доп. М.: Наука, 1967. - 367 с.

Практичне заняття № 2

Тема: Виконання найпростіших арифметичних операцій.

Мета: Оволодіти логічним базисом, що представляє собою сукупність типів найпростіших арифметичних операцій.

Завдання

1. Скласти два додатніх числа $a = 1001_2$ і $b = 1101_2$.
2. Скласти два десяткових числа $a = +831_2$ і $b = -376_2$ з використанням додаткового коду.
3. Скласти два числа $a = \bar{0}10110_2$ і $b = \bar{1}01101_2$.
4. Обчислити суму $a = \bar{1}10110_2$ і $b = \bar{0}01101_2$.
5. Обчислити суму $a = +44_{10}$ і $b = -96_{10}$.
6. Виконати додавання двох чисел $a = 39_{10}$ і $b = 13_{10}$ у рамках розрядної сітки, що дорівнює шести розрядам.
7. Виконати додавання двох чисел $a = 52_{10}$ і $b = 13_{10}$ у рамках розрядної сітки, що дорівнює шести розрядам.
8. Виконати додавання двох чисел 999 і -999, рівних за модулем та протилежних за знаком.
9. Скласти два довільно вибрані від'ємні числа.
10. Виконати додавання двох чисел $a = 0,111_2$ і $b = 0,101_2$.
11. Визначити абсолютне значення добутку двох чисел $a = 1101_2$ і $b = 1011_2$.

Теоретичні відомості

Основною операцією, яка використовується в цифрових системах при виконанні різних обчислень, є операція *алгебраїчного додавання*. Вона виконується на основі правил виконання операцій у двійковій системі зображення чисел, які для однорозрядних чисел мають такий вигляд:

$$\begin{array}{r} + \quad 0 \\ \quad 0 \\ \hline 0 \end{array} \quad \begin{array}{r} + \quad 1 \\ \quad 0 \\ \hline 1 \end{array} \quad \begin{array}{r} + \quad 0 \\ \quad 1 \\ \hline 1 \end{array} \quad \begin{array}{r} + \quad 1 \\ \quad 1 \\ \hline 10 \end{array}$$

Перенесення до старшого розряду виконується тоді, коли в одному розряді обох складових є одиниці. Операція знаходження суми в багаторозрядних числах виконується послідовно, починаючи з молодшого розряду. У зв'язку з цим, починаючи з другого розряду, виконується складання трьох цифр – двох розрядних складових і перенесення з молодшого розряду.

Операція *віднімання* в цифрових схемах виконується за допомогою операції додавання із зображенням від'ємника у доповнюючому коді. Про доповнюючий код нічого не сказано

Пряма операція віднімання з використанням операцій зайому зі старших розрядів застосовується лише при порівнянні двох кодів, адже відсутність чи наявність зайому зі старшого розряду дає можливість визначити, яке з порівнюваних чисел більше.

Для довільних чисел виконання операції множення зводиться до послідовного виконання операцій додавання та зсуву.

Контрольні запитання

1. Назвіть основні логічні функції.
2. Поясніть послідовність виконання арифметичних операцій додавання і віднімання в двійковій системі числення при різних знаках зменшуваного і від'ємника.

3. Поясніть особливість виконання арифметичних операцій у двійково-десятковому коді.

4. Поясніть послідовність виконання операції множення у двійковому коді.

5. Операція віднімання у двійковому коді виконується з використанням доповнюючого коду. Проаналізуйте і дайте пояснення, чи справедлива подібна послідовність виконання арифметичної операції, якщо числа задаються у двійково-десятковому коді; у коді «з надлишком 3».

Рекомендована література

1. J. Eldon Whitesitt Boolean Algebra and Its Applications. Courier Corporation, 1995. 182 pp.

2. Браммер Ю.А. Цифрові пристрої: Учеб. посібник для вузів. -М.: Вищ. шк., 2004. -229 с.

3. Пухальський Г.І., Новосельцева Т.Я. Цифрові пристрої: Учеб. посібник для втузів .- СПб.: Політехніка, 1996 .- 885 с.

4. Угрюмов Є.П. Цифрова схемотехніка: Учеб. посібник для вузов.-СПб: БХВ-Петербург, 2000, 2004. – 528 с.

Практичне заняття № 3 – 4

Тема: Алгебра логіки.

Мета: Вивчення основних законів алгебри логіки та їх доведення з використанням логічних елементів «АБО», «ВИКЛЮЧАЮЧЕ АБО», «І» та «ВИКЛЮЧАЮЧЕ І».

Завдання

1. Скласти таблиці істинності для двовходових елементів «АБО», «І», «ВИКЛЮЧАЮЧЕ АБО».
2. Привести карти Карно для вказаних у завданні № 1 логічних функцій.
3. Побудувати часові імпульсні діаграми, що ілюструють роботу двовходових логічних елементів «АБО», «ВИКЛЮЧАЮЧЕ АБО», «І» та «ВИКЛЮЧАЮЧЕ І».
4. Представити за допомогою карт Карно наступні функції трьох змінних:
 $y_1 = x_1 x_2 x_3 + \bar{x}_1 \bar{x}_2 \bar{x}_3$, $y_2 = x_2 x_3$, $y_3 = \bar{x}_3$.
5. Представити за допомогою карт Карно наступні функції чотирьох змінних: $y_1 = \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4$, $y_2 = x_1 x_3 x_4$, $y_3 = \bar{x}_2 x_4$, $y_4 = \bar{x}_2$.
6. Мінімізувати логічні функції, представлені картами Карно (рис. 1 а, б, в), та реалізувати їх у базисі І, АБО, НІ.

| x_1, x_2 x_3, x_4 | | x_1, x_2 | | | |
|--------------------------|----|------------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| 00 | 00 | 1 | 0 | 0 | 1 |
| 01 | 01 | 0 | 1 | 1 | 0 |
| 11 | 11 | 0 | 1 | 1 | 0 |
| 10 | 10 | 1 | 0 | 0 | 1 |

а

| x_1, x_2 x_3, x_4 | | x_1, x_2 | | | |
|--------------------------|----|------------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| 00 | 00 | 0 | 1 | 1 | 0 |
| 01 | 01 | 1 | 0 | 0 | 1 |
| 11 | 11 | 1 | 0 | 0 | 1 |
| 10 | 10 | 0 | 1 | 1 | 0 |

б

Рис. 1

| x_1, x_2 x_3, x_4 | | x_1, x_2 | | | |
|--------------------------|----|------------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| 00 | 00 | 1 | 0 | 0 | 1 |
| 01 | 01 | 0 | 1 | 1 | 0 |
| 11 | 11 | 1 | 0 | 0 | 1 |
| 10 | 10 | 1 | 0 | 0 | 1 |

в

7. Таблицями істинності (табл.1) задані три логічні функції трьох змінних. Представити їх в аналітичній формі та мінімізувати. Реалізувати мінімізовані функції у базисі І, АБО, НІ.

Таблиця 1

| X ₃ | X ₂ | X ₁ | Y ₁ | Y ₂ | Y ₃ |
|----------------|----------------|----------------|----------------|----------------|----------------|
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |

8. Побудувати таблиці істинності для наступних функцій трьох змінних:

- 1) сума за модулем два;
- 2) нерівність всіх аргументів один одному;
- 3) **ВИКЛЮЧАЮЧЕ АБО** – альтернатива, тобто один і тільки один;
- 4) диз'юнкція.

9. Мінімізувати логічну функцію:

$$y = x_1x_2x_3 + \bar{x}_1x_2x_3 + \bar{x}_1\bar{x}_2x_3 + \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1x_2\bar{x}_3.$$

10. Довести тотожності:

- 1) $(x_1 + x_3) \cdot x_2 = x_1x_2\bar{x}_3 + \bar{x}_1x_2x_3 + x_1x_2x_3;$
- 2) $x_1x_3 + x_2\bar{x}_3 + x_1x_2 = x_1x_3 + x_2\bar{x}_3;$
- 3) $x_2x_3 + x_1x_4 = (x_1 + x_2) \cdot (x_2 + x_4) \cdot (x_1 + x_3) \cdot (x_3 + x_4);$
- 4) $x_1x_2 + x_1x_2x_3 + x_1\bar{x}_2 = x_1;$
- 5) $\bar{x}_1x_2 + x_1\bar{x}_2 = (\bar{x}_1 + \bar{x}_2) \cdot (x_1 + x_2).$

11. Знайти заперечення для кожного із наступних виразів. Результати не спрощувати.

- 1) $y_1 = x_1 + \bar{x}_2x_3;$
- 2) $y_2 = x_1 \cdot (x_2 + x_3) + x_2\bar{x}_4 \cdot (\bar{x}_1 + x_3);$
- 3) $y_3 = [x_1\bar{x}_2 + x_3 \cdot (\bar{x}_1 + x_4x_5)] \cdot [\bar{x}_2 + x_1x_3 \cdot (\bar{x}_4 + \bar{x}_2\bar{x}_3)];$
- 4) $y_4 = \overline{x_1x_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3} + \overline{(x_1 + x_2 + x_3) \cdot (\bar{x}_1 + \bar{x}_2 + x_3)};$

$$5) y_5 = \overline{\overline{(x_1 + x_2 \bar{x}_3)} \cdot (\bar{x}_2 x_3 + \bar{x}_1)} + \overline{(x_1 x_2 \bar{x}_3 + \bar{x}_2)}.$$

12. Скласти таблицю істинності для наступних функцій:

$$1) y_1 = x_1 + x_2 \bar{x}_3;$$

$$2) y_2 = x_2 x_4 + \bar{x}_1 x_3 + \bar{x}_2 \bar{x}_4;$$

$$3) y_3 = x_1 x_3 + x_2 x_3 + x_1 x_2$$

$$4) y_4 = x_2 \bar{x}_4 + x_1 \bar{x}_3 + \bar{x}_2 \bar{x}_4.$$

13. Використовуючи теорему де-Моргана, перетворити наступні функції:

$$1) y_1 = \overline{x_1 + x_2} + \overline{x_1 x_2 x_3} + \overline{\bar{x}_1 x_3};$$

$$2) y_2 = \overline{(x_1 x_2 + \bar{x}_2 x_3)} + \overline{(x_2 \bar{x}_3 + \bar{x}_1 x_2)};$$

$$3) y_3 = \overline{(x_1 x_2 + \bar{x}_2 x_3)} \cdot \overline{(x_2 x_3 + \bar{x}_1 \bar{x}_3)};$$

$$4) y_4 = \overline{\overline{(x_1 + x_2)} + (x_1 + x_2)} \cdot \overline{\overline{(\bar{x}_3 + \bar{x}_2)} + (\bar{x}_3 + \bar{x}_2)};$$

$$5) y_5 = \overline{\overline{(x_1 + \bar{x}_2 x_3 + \bar{x}_1 \bar{x}_2)}} + \overline{\overline{(x_1 + x_2 \bar{x}_3 \bar{x}_1)}};$$

$$6) y_6 = \overline{x_1 + \bar{x}_2 \bar{x}_3} + \overline{\bar{x}_1 x_2 x_3} + \overline{\bar{x}_2 x_3}.$$

14. Мінімізувати логічну функцію за допомогою карти Карно:

$$y = x_2 x_3 x_4 + \bar{x}_1 x_2 x_4 + \bar{x}_2 x_3 x_4 + \bar{x}_1 x_2 x_3 + x_1 \bar{x}_3 x_4 + \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_2 \bar{x}_3 \bar{x}_4.$$

15. Мінімізувати за допомогою карт Карно наступні функції:

$$1) y_1 = f(x_1, x_2, x_3, x_4) = \sum 0, 1, 2, 3, 4, 6, 7, 8, 12, 13;$$

$$2) y_2 = f(x_1, x_2, x_3) = \sum 0, 1, 2, 3, 7;$$

$$3) y_3 = f(x_1, x_2, x_3) = \sum 3, 5, 6, 7.$$

16. Представити у мінімальній диз'юнктивній формі наступні функції:

$$1) y_1 = x_1(x_1 \oplus x_2 \oplus x_3);$$

$$2) y_2 = x_1(x_1 \otimes x_2 \otimes x_3);$$

$$3) y_3 = x_1 + (x_1 \oplus x_2 \oplus x_3);$$

$$4) y_4 = x_1 + (x_1 \otimes x_2 \otimes x_3).$$

17. Довести тотожності:

- 1) $\overline{x_1 \oplus x_2 \oplus x_3} = x_1 \oplus x_2 \otimes x_3$;
- 2) $x_1 \otimes x_2 \otimes x_3 = x_1 \otimes x_2 \oplus x_3$;
- 3) $(x_1 \oplus x_2 \oplus x_1 x_2) \cdot (x_1 \oplus x_2 \oplus x_1 x_3) = x_1 + x_2 x_3$.

18. Мінімізувати функції та реалізувати їх у базисі І-НІ:

- 1) $y_1 = f(x_1, x_2, x_3) = \sum 0, 1, 2, 3, 4, 5, 6$;
- 2) $y_2 = f(x_1, x_2, x_3, x_4) = \sum 1, 3, 4, 5, 7, 9, 11$.

19. Використовуючи елементи І-НІ, реалізовувати наступні функції, не застосовуючи прийоми мінімізації:

- 1) $y_1 = x_1 \bar{x}_2 + (\bar{x}_2 + \bar{x}_3) \cdot x_1$;
- 2) $y_2 = (x_1 x_2 + x_3) \cdot (x_3 + \bar{x}_4) + x_1 (\bar{x}_2 + x_3) \cdot (x_4 + \bar{x}_5)$.

20. Реалізувати наступну функцію на елементах І-НІ, що мають не більше трьох входів:

$$y = x_1 x_2 \bar{x}_3 + x_1 x_4 + x_2 \bar{x}_3 \bar{x}_4 + x_1 x_2 \bar{x}_4.$$

21. Використовуючи елементи АБО-НІ, реалізовувати наступні булеві функції:

- 1) $y_1 = x_1 (\bar{x}_1 + \bar{x}_2) \cdot (x_2 + \bar{x}_3 x_4)$;
- 2) $y_2 = x_1 \cdot (x_2 + x_3 + x_4 x_5) \cdot (\bar{x}_2 + x_3 x_4 + \bar{x}_1 x_5)$;
- 3) $y_3 = (x_1 \bar{x}_3 + x_2 x_3) \cdot (\bar{x}_1 + \bar{x}_3)$.

22. Реалізувати на елементах АБО-НІ, що мають не більше трьох входів, функцію $y = \bar{x}_1 x_2 + x_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 \bar{x}_4$.

23. Реалізувати логічну схему пристрою з непарною кількістю входів m , який дозволяє вирішити задачу голосування пристрою більшістю. Розв'язати задачу для $m=3$ та $m=5$.

24. При розробці електронного календаря виникає необхідність кодування днів тижня. Пропонується довільно закодувати ді тижня та розробити логічну схему пристрою кодування.

25. Для логічної функції, що задана картою Карно (рис. 2), значення функції у клітинках зі значком «*» є недовизначеними. Довизначити функцію з метою отримання мінімальної диз'юнктивної форми.

| $x_3, x_4 \backslash x_1, x_2$ | 00 | 01 | 11 | 10 |
|--------------------------------|----|----|----|----|
| 00 | 1 | 1 | 0 | * |
| 01 | * | * | 0 | * |
| 11 | 1 | 1 | * | 1 |
| 10 | 1 | 1 | * | 1 |

Рис. 2

26. Недовизначену функцію, що задана картою Карно (рис. 2), записати в мінімальній кон'юнктивній формі.

27. Визначити величину часової затримки трьох послівно включених мікросхем К155ЛА3 та порівняти з результатами затримок аналогічних мікросхем серій К555, 1533, К561.

28. На логічних мікросхемах серії К155 побудувати декілька варіантів схем, що реалізують логічну функцію $y = \overline{x_1 + x_3 + x_2 + x_3 + \bar{x}_4}$.

29. Використовуючи мікросхеми з відкритим колектором, розробити схеми «монтажне АБО», «монтажне І».

30. На рис. 3 приведені логічні схеми монтажно́ї логіки. Пояснити, які логічні операції вони реалізують.

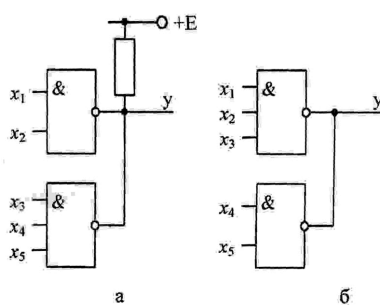


Рис. 3

31. Забезпечити розгалудження сигналу на число елементів, що перевищує коефіцієнт розгалудження по виходу елементу, що служить джерелом сигналу. Дати розв'язки як для невеликої кратності перевищення, так і перевищення в десятки разів.

32. На елементах серії K155 побудувати схеми елементів, які реалізують операції $16 - I$ та $16 - ABO$.

33. Побудувати схему пристрою додавання за модулем 2 на вісім входів, використовуючи двовходові суматори за модулем 2.

34. Побудувати схему вузла мажорювання 2 з 3, що відновлює інформацію при відмові одного з трьох каналів. Побудувати вузол, що дозволяє виявити канал, який відмовив.

Теоретичні відомості

Математичний апарат, що оперує з аргументами та функціями, які набувають тільки двох значень – “0” та “1” – називається *двійковою (булевою) алгеброю* або *алгеброю логіки*.

Якщо змінна x_i набуває значення $x_i = 1$, то таке її значення називають *істинним*. Протилежне $x_i = 0$ називають *хибним* і умовно позначають $\overline{x_i}$, що означає заперечення істинного значення аргументу. Два елементи булевої алгебри – подія істинна і подія хибна – називають її *константами*.

Булева функція позначається літерою y і є двійковою функцією двійкових аргументів. Умовне її позначення

$$y = f(x_1, x_2, \dots, x_n).$$

Функція повністю визначена, якщо задані її значення в усіх точках області визначеності. Якщо ж значення функції не задано в одній або кількох точках, то вона є *неповністю визначеною*.

Усі можливі логічні функції n змінних можна створити за допомогою трьох основних операцій:

а) логічне заперечення (інверсія, операція **НІ**); позначається рискою над відповідною функцією або аргументом;

б) логічне додавання (диз'юнкція, операція **АБО**), яке позначається символами (V), (+);

в) логічне множення (кон'юнкція, операція \mathbf{I}), яке позначається символами (\wedge) , (\cdot) , $(\&)$. Для позначення еквівалентності логічних виразів використовується знак $(=)$.

Закони і тотожності алгебри логіки. Основні аксіоми та закони булевої алгебри (*переміщувальний (властивість комутативності); сполучний (властивість асоціативності); розподільний (властивість дистрибутивності); інверсії (теорема де Моргана)*) наведені у табл. 2.

Таблиця 2

| Назва аксіоми чи закону | Вирази |
|---|--|
| Аксіоми (тотожності) | $0 \cdot x = 0$ $1 + x = 1$ $0 + x = x$ $x \cdot x = x$ $x + x = x$ $x \cdot \bar{x} = 0$ $x + \bar{x} = 1$ $\bar{\bar{x}} = x$ |
| Закони комутативності | $x_1 + x_2 = x_2 + x_1$ $x_1 \cdot x_2 = x_2 \cdot x_1$ |
| Закони асоціативності | $x_1 + x_2 + x_3 = x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3 =$ $= (x_1 + x_3) + x_2$ $x_1 \cdot x_2 \cdot x_3 = x_1 \cdot (x_2 \cdot x_3) =$ $= x_2 \cdot (x_1 \cdot x_3) = x_3 \cdot (x_1 \cdot x_2)$ |
| Закони дистрибутивності | $x_1 \cdot (x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3$ $x_1 + x_2 \cdot x_3 = (x_1 + x_2) \cdot (x_1 + x_3)$ |
| Закони інверсії (теорема де Моргана, принцип подвійності) | $\overline{x_1 \cdot x_2} = \bar{x}_1 + \bar{x}_2$ $\overline{x_1 + x_2} = \bar{x}_1 \cdot \bar{x}_2$ |
| Закони поглинання | $x_1 + x_1 \cdot x_2 = x_1$ $x_1 \cdot (x_1 + x_2) = x_1$ |

Способи задання логічних функцій. Аналітично логічна функція може бути записана різними комбінаціями кон'юнкцій та диз'юнкцій логічних змінних. Зазвичай логічні функції записуються або у вигляді суми добутків логічних змінних (диз'юнкція кон'юнкцій) або у вигляді логічного добутку їх сум (кон'юнкція диз'юнкцій). Наведення функції у вигляді диз'юнкції кон'юнкцій називають *диз'юнктивною нормальною формою (ДНФ)*, а запис у вигляді кон'юнкції диз'юнкцій – відповідно, *кон'юнктивною нормальною формою (КНФ)*. Інверсія у відповідності з теоремою де Моргана будь-якої функції, приведений в одній формі, призводить до заміни запису на іншу форму. Існує лише один вид **ДНФ** та **КНФ**, в яких функція може бути записана єдиним чином. Такі форми називаються *досконалими диз'юнктивними (кон'юнктивними) нормальними формами (ДДНФ, ДКНФ)*. Вони характеризуються тим, що в **ДДНФ** кожна кон'юнкція, а в **ДКНФ** кожна диз'юнкція містять усі логічні змінні даної функції, з інверсіями або без них. Оскільки кожна кон'юнкція функції, що наведена у **ДДНФ**, визначає її істинне значення, відповідаюче “1”, то такі кон'юнкції називаються *конституєнтами одиниці (мінтермами)*. Аналогічно, диз'юнкції функції, що наведені у **ДКНФ**, називаються *конституєнтами нуля (макстермами)*.

Досконала диз'юнктивна нормальна форма запису дозволяє легко перейти до інших форм запису – *табличної та карт Карно*.

Таблиця 3.

| x_1 | x_0 | y_1 | y_2 | y_3 | y_4 | y_5 |
|-------|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | x | 1 | 0 | 1 | 1 |

У табл. 3 наведені функції $y_1 \dots y_5$ двох змінних x_0 та x_1 . Табличний спосіб полягає у тому, що функція задається у вигляді таблиці відповідності (таблиці істинності станів). У таблицю вписують усі можливі комбінації аргументів у порядку зростання їх індексів і при кожній комбінації встановлюється значення функції. Кількість всіх можливих сполук аргументів, а, отже, і кількість значень функції дорівнює 2^n , де n – кількість логічних змінних. З табличної форми запису легко перейти до

аналітичної, використовуючи досконалу диз'юнктивну форму запису логічних функцій. Для цього функція записується як диз'юнкція конституент одиниці.

У табл. 4 наведені всі елементарні функції двох змінних.

Таблиця 4

| N | x_0 | 0 | 1 | 0 | 1 | Назва функції | Позначення |
|-----|-------|---|---|---|---|---------------------------------|--|
| | x_1 | 0 | 0 | 1 | 1 | | |
| 0 | | 0 | 0 | 0 | 0 | Константа нуль | 0 |
| 1 | | 0 | 0 | 0 | 1 | Кон'юнкція, I | $x_1 \cdot x_0$ |
| 2 | | 0 | 0 | 1 | 0 | Заборона по x_0 | $x_1 \cdot \overline{x_0}$ |
| 3 | | 0 | 0 | 1 | 1 | Змінна x_1 | x_1 |
| 4 | | 0 | 1 | 0 | 0 | Заборона по x_1 | $\overline{x_1} \cdot x_0$ |
| 5 | | 0 | 1 | 0 | 1 | Змінна x_0 | x_0 |
| 6 | | 0 | 1 | 1 | 0 | Викл. АБО, сума за mod 2 | $x_1 \oplus x_0$ |
| 7 | | 0 | 1 | 1 | 1 | Диз'юнкція, АБО | $x_1 + x_0$ |
| 8 | | 1 | 0 | 0 | 0 | АБО-НІ, функція Пірса | $\overline{x_1 + x_0}$ |
| 9 | | 1 | 0 | 0 | 1 | Рівнозначність, еквівалентність | $x_1 \equiv x_0$ |
| 10 | | 1 | 0 | 1 | 0 | Заперечення x_0 | $\overline{x_0}$ |
| 11 | | 1 | 0 | 1 | 1 | Імплікація по x_0 | $\overline{x_1 + x_0} \cdot x_0 \rightarrow x_1$ |
| 12 | | 1 | 1 | 0 | 0 | Заперечення x_1 | $\overline{x_1}$ |
| 13 | | 1 | 1 | 0 | 1 | Імплікація по x_1 | $\overline{x_1 \rightarrow x_0} \cdot x_1 + x_0$ |
| 14 | | 1 | 1 | 1 | 0 | Функція Шеффера I-НІ | $\overline{x_1 x_0}$ |
| 15 | | 1 | 1 | 1 | 1 | Константа 1 | 1 |

Карта Карно – це компактна форма представлення таблиці істинності логічної функції із застосуванням для позначення (кодування) комбінацій змінних циклічного коду Грея. Особливість карти Карно полягає в тому, що по горизонталі та по вертикалі задаються координати клітинок, якими виступають аргументи логічної функції. Тому кожна клітина має свою координату – 00, 01, 10, 11 – яка може бути представлена відповідною двійковою або десятковою цифрою. Значення функції записуються в відповідних клітинах нулями та одиницями.

При використанні карт Карно для функцій з більшою кількістю логічних змінних кількість клітин у карті зростає відповідно до формули

$$N = 2^n.$$

Якщо задана функція неповністю визначена, то, подібно до таблиці, у відповідних клітинах записується знак невизначеності, що дає можливість повністю відобразити всі властивості функції.

Графічна інтерпретація ґрунтується на використанні умовних позначень логічних елементів, які є стандартними.

На рис. 4 приведені умовні позначення, характерні для стандартів, прийнятих у країнах СНД.

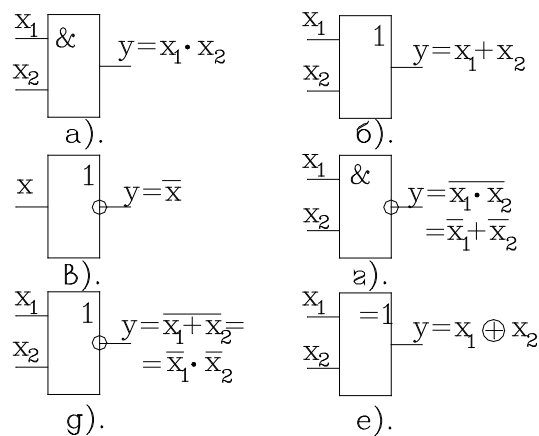


Рис. 4

Мінімізація логічних функцій. Метою мінімізації є зменшення вартості технічної реалізації логічних функцій незалежно від використовуваних апаратних засобів. Складність логічних функцій визначається кількістю логічних змінних, що входять до їх складу в прямому і інверсному виді, та кількістю простих логічних операцій над ними. Будь-яка логічна функція може бути записана різними аналітичними виразами різного рівня складності. Серед них можна знайти такі, які містять мінімальну кількість логічних змінних і операцій над ними. Задача знаходження таких аналітичних виразів називається *мінімізацією логічних функцій*. Звідси витікає, що *мінімізація логічної функції* – це заміна логічної функції, що представлена у вигляді логічної суми мінтермів

або логічного добутку макстермів, іншою логічною функцією з мінімальною кількістю логічних змінних та операцій над ними.

Аналітичний спосіб мінімізації. Для зменшення складності логічних функцій здебільшого використовуються операції склеювання:

$$\begin{aligned}x_1 x_2 + x_1 \overline{x_2} &= x_1 (x_2 + \overline{x_2}) = x_1 ; \\(x_1 + x_2) (x_1 + \overline{x_2}) &= x_1\end{aligned}$$

та поглинання:

$$\begin{aligned}x_1 + x_1 x_2 &= x_1 (1 + x_2) = x_1 ; \\x_1 (x_1 + x_2) &= x_1 .\end{aligned}$$

Як приклад, розглянемо процедуру спрощення наступної функції:

$$\begin{aligned}y &= x_1 x_2 x_3 + \overline{x_1} x_2 x_3 + \overline{x_1} x_3 \overline{x_2} + \overline{x_1} \overline{x_2} \overline{x_3} + \overline{x_1} x_2 \overline{x_3} = \\&= x_1 x_2 x_3 + (\overline{x_1} x_2 + \overline{x_1} \overline{x_2}) (x_3 + \overline{x_3}) = x_1 x_2 x_3 + \overline{x_1} x_2 + \overline{x_1} \overline{x_2} = \\&= x_1 x_2 x_3 + \overline{x_1} (x_2 + \overline{x_2}) = x_1 x_2 x_3 + \overline{x_1} .\end{aligned}$$

Одержана ДНФ має мінімальну складність.

Мінімізація за допомогою карт Карно (діаграм Вейча). Метод, оснований на використанні карт Карно, характеризується своєю простотою і наочністю. Основні властивості карт Карно:

- клітини карти, координати яких відрізняються лише параметрами однієї змінної, називаються *сусідніми*;
- сусідні клітини, значення функцій в яких або тільки істинні, або тільки хибні, можуть об'єднуватися в групи по 2^m клітин, де m – ціле число ($m = 0, 1, 2, 3, \dots$);
- при переході до аналітичної форми запису логічної функції з карти Карно вона може записуватись незмінними координатами об'єднаних груп клітин;
- у випадку неповністю визначеної функції невизначені клітини можуть бути довизначеними, виходячи з умови одержання більшої кількості об'єднаних клітин;
- одна клітина може об'єднуватись у декілька груп.

Використання карт Карно для мінімізації логічних функцій базується на наглядному використанні операції склеювання. Дійсно, дві сусідні клітки відрізняються лише однією змінною. Тому, об'єднуючи їх, ми записуємо лише незмінні координати, тобто виносимо їх за дужки. У дужках залишаються змінні координати, які об'єднуються в одну операцією склеювання. Для отримання мінімального значення функції, представленого картою Карно, окрім правил, викладених вище, необхідно користуватись загальним правилом: одиниці або нулі повинні об'єднуватись мінімальним числом найбільших контурів.

За допомогою карт Карно легко вирішуються задачі мінімізації функцій з кількістю змінних до шести включно. При більшій кількості змінних пошук мінімальних форм запису функцій значно ускладнюється і наочність карт Карно втрачається.

Контрольні запитання

1. Дайте визначення терміну “алгебра логіки” (“булева алгебра”).
2. У чому полягає суть принципу подвійності?
3. Дайте пояснення диз'юнктивній і кон'юнктивній формам запису логічних функцій, а також досконалим формам запису.
4. Перелічіть відомі Вам способи запису логічних функцій. Дайте пояснення взаємозв'язку між ними.
5. Яка функція називається *неповністю визначеною*? Як забезпечити перехід для неповністю визначеної функції від табличного способу запису до аналітичного?
6. Дайте пояснення суті теореми де Моргана. Приведіть приклади її використання.
7. Поясніть властивості карти Карно.

8. Які логічні операції використовуються для аналітичного способу мінімізації логічних функцій?
9. На яких властивостях карт Карно реалізується задача мінімізації логічних функцій?
10. На якій концепції базується мінімізація з використанням кубічних комплексів?
11. Поясніть особливість мінімізації неповністю визначених функцій.
12. У чому полягає особливість сумісної мінімізації декількох логічних функцій?

Рекомендована література

1. J. Eldon Whitesitt Boolean Algebra and Its Applications. Courier Corporation, 1995. 182 pp.
2. Новіков Ю.В. Основи цифрової схемотехніки. Базові елементи і схеми. Методи проектування. М.: Світ, 2001. - 379 с.
3. Новіков Ю.В., Скоробогатов П.К. Основи мікропроцесорної техніки. Курс лекцій. М.: ІНТУІТ.РУ, 2003. - 440 с.
4. Пухальський Г.І., Новосельцева Т.Я. Цифрові пристрої: Учеб. посібник для Втузов. СПб.: Політехніка, 2006. - 885 с.
5. Преснухін Л.М., Воробйов Н.В., Шішкевіч А.А. Розрахунок елементів цифрових пристроїв. М.: Вищ. шк., 2001. - 526 с.
6. Букреєв І.М., Горячев В.І., Мансуров Б.М. Мікроелектронні схеми цифрових пристроїв. М.: Радіо і зв'язок, 2000. - 416 с.
7. Соломатін Н.М. Логічні елементи ЕОМ. М.: Вищ. шк., 2000. - 160 с.

Практичне заняття № 5

Тема: Мінімізація логічних функцій методом Квайна.

Мета роботи: Вивчення методів сумісної мінімізації систем логічних функцій; аналізу і синтезу комбінаційних логічних схем з багатьма виходами.

Завдання

1. Спростити методом Квайна логічну функцію задану таблицею.

| | | | | | | | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| x ₁ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| x ₂ | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| x ₃ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| x ₄ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| y | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |

Теоретичні відомості

Сумісна мінімізація систем логічних функцій потрібна при проектуванні комбінаційних схем з багатьма виходами. Саме такі схеми описуються системою логічних функцій, число яких дорівнює числу виходів схеми.

Задача сумісної мінімізації системи **m** логічних функцій від **n** змінних полягає у виявленні та раціональному використанні *спільних логічних виразів*. Якщо виконувати незалежну мінімізацію окремо кожної функції системи, то хоч кожна з цих функцій буде мінімальною, але в цілому система найчастіше виявляється *немінімальною*.

Відомо кілька способів сумісної мінімізації. Ці способи ділять на дві групи. До першої групи відносять методи, що використовують нормальні (дворівневі) канонічні форми функцій. До другої групи відносять методи, що використовують багаторівневі представлення функцій.

Найпоширеніший метод сумісної мінімізації *першої групи* ґрунтується на знаходженні простих імплікант, якими покривається кожна функція заданої системи. Цей метод нагадує *метод Квайна* і може бути описаний наступним *алгоритмом*:

1. Записати кожен функцію заданої системи в досконалій диз'юнктивній нормальній формі (ДДНФ) і сформувані повну множину **A** мінтермів, які відповідають одиничним значенням всіх функцій системи. Кожному мінтерму множини **A** приписати ознаку входження в ДДНФ тої чи іншої функції системи.

2. Здійснити мінімізацію допоміжної функції **Z**, ДДНФ якої містить *всі* мінтерми множини **A**. В процесі мінімізації, виконуючи склеювання мінтермів або імплікант, результату склеювання присвоїти ознаку, що складається з номерів функцій, *спільних* для мінтермів або імплікант, що склеюються. Якщо ознаки мінтермів або імплікант не містять однакових номерів, склеювання *не здійснюється*. *Поглинання* здійснюється тільки для кон'юнкцій з однаковими ознаками. Отримані внаслідок склеювання і поглинання кон'юнкції називають *простими імплікантами системи функцій*.

3. Для допоміжної функції **Z** побудувати імплікантну таблицю, подібну до *матриці Квайна*, але для кожного мінтерма виділити стільки стовпців, скільки різних номерів функцій містить ознака цього мінтерма. Покриття мінтермів імплікантами здійснюється за методом Квайна.

Розглянемо реалізацію описаного алгоритму на прикладі.

Приклад 1. Виконати сумісну мінімізацію системи логічних функцій, заданої таблицею істинності (табл. 5). Побудувати комбінаційну схему для реалізації заданої системи функцій, використовуючи елементи І, АБО, НЕ.

Таблиця 5

| № набору | x_1 | x_2 | x_3 | y_1 | y_2 | y_3 |
|-------------|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 | 0 | 1 |
| 4 | 1 | 0 | 0 | 0 | 1 | 0 |
| 5 | 1 | 0 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 |

Розв'язання.

1. Перший крок при реалізації даного методу мінімізації - запис кожної з функцій системи в ДДНФ, тобто у вигляді диз'юнкції мінтермів, що відповідають одиничним значенням логічних функцій. Отже, спираючись на Таблицю 5, запишемо ДДНФ трьох заданих логічних функцій:

$$\begin{cases} y_1 = \overline{x_1} \overline{x_2} \overline{x_3} + \overline{x_1} \overline{x_2} x_3 + \overline{x_1} x_2 \overline{x_3} + \overline{x_1} x_2 x_3 \\ y_2 = \overline{x_1} \overline{x_2} x_3 + \overline{x_1} \overline{x_2} \overline{x_3} + \overline{x_1} x_2 \overline{x_3} + \overline{x_1} x_2 x_3 \\ y_3 = \overline{x_1} \overline{x_2} \overline{x_3} + \overline{x_1} \overline{x_2} x_3 + \overline{x_1} x_2 \overline{x_3} + \overline{x_1} x_2 x_3 \end{cases} \quad (5.1)$$

Наступний крок: формуємо множину **A** - з системи (5.1) виписуємо всі різні мінтерми, приписуючи кожному ознаку входження в функцію y_1 , y_2 , чи y_3 . Тут під ознакою маємо на увазі сукупність номерів функцій, ДДНФ яких містить даний мінтерм:

$$\overline{x_1} \overline{x_2} \overline{x_3} \rightarrow y_3 \Rightarrow (\overline{x_1} \overline{x_2} \overline{x_3})_3$$

$$\overline{x_1} \overline{x_2} x_3 \rightarrow y_2, y_3 \Rightarrow (\overline{x_1} \overline{x_2} x_3)_{2,3}$$

$$\overline{x_1} x_2 \overline{x_3} \rightarrow y_1 \Rightarrow (\overline{x_1} x_2 \overline{x_3})_1$$

$$\overline{x_1} x_2 x_3 \rightarrow y_1, y_3 \Rightarrow (\overline{x_1} x_2 x_3)_{1,3}$$

$$x_1 \overline{x_2} \overline{x_3} \rightarrow y_2 \Rightarrow (x_1 \overline{x_2} \overline{x_3})_2$$

$$x_1 \overline{x_2} x_3 \rightarrow y_2 \Rightarrow (x_1 \overline{x_2} x_3)_2$$

$$x_1 x_2 \overline{x_3} \rightarrow y_1, y_3 \Rightarrow (x_1 x_2 \overline{x_3})_{1,3}$$

$$x_1 x_2 x_3 \rightarrow y_1, y_2, y_3 \Rightarrow (x_1 x_2 x_3)_{1,2,3}$$

З отриманої множини мінтермів **A** будуюмо ДДНФ допоміжної функції **Z** (вказуємо при цьому ознаки):

$$Z = (\overline{x_1} \overline{x_2} \overline{x_3})_3 + (\overline{x_1} \overline{x_2} x_3)_{2,3} + (\overline{x_1} x_2 \overline{x_3})_1 + (\overline{x_1} x_2 x_3)_{1,3} + (x_1 \overline{x_2} \overline{x_3})_2 + (x_1 \overline{x_2} x_3)_2 + (x_1 x_2 \overline{x_3})_{1,3} + (x_1 x_2 x_3)_{1,2,3} \quad (5.2)$$

2. Далі приступаємо до знаходження простих імплікант за Квайном. Перед цим для зручності *пронумеруємо* кожний мінтерм функції **Z** (див. (5.2)).

Знаходження простих імплікант за Квайном зводиться до пошуку пар мінтермів та імплікант, що склеюються між собою. При цьому слід розрізняти (і не склеювати між собою) мінтерми, що входять до різних функцій, тобто такі мінтерми, ознаки яких не містять однакових номерів функцій.

Зауваження: процес пошуку простих імплікант є багатостадійним, причому максимальна кількість цих етапів на одиницю менша за кількість логічних змінних, від яких залежать функції системи. На першому етапі здійснюють всі можливі склеювання мінтермів функції **Z**. На другому етапі здійснюють склеювання імплікант, що утворилися як результат виконання першого етапу, тобто склеюють кон'юнкції, кількість змінних в яких на одиницю менша, ніж в мінтермах. І так далі, аж поки на останньому етапі не розглянуть можливість склеювання імплікант, що містять тільки дві змінні.

Для нашого прикладу цей процес має два етапи (три логічні змінні), і поетапно буде виглядати так:

1-ий етап: на цьому етапі здійснюємо склеювання мінтермів функції **Z**.

Етап можна умовно поділити на два кроки:

1) склеювання - в (2) відшукуємо всі пари мінтермів, що склеюються (мінтерми таких пар крім звичайних властивостей повинні мати в ознаці хоча б один спільний номер функції). При склеюванні результат записуємо як диз'юнкцію імпліканти і (не завжди) мінтермів, що склеювалися. Імпліканті присвоюємо ознаку, яка містить номери функцій, спільні для ознак обох мінтермів пари. Мінтерм, що брав участь в склеюванні, входить до результату склеювання, якщо його ознака не збігається повністю з ознакою імпліканти. Мінтерм, ознака якого повністю збігається з ознакою імпліканти, до результату склеювання не входить (він *поглинається* імплікантою). У відповідності з викладеним для нашого прикладу отримаємо (зліва в дужках вказано номери мінтермів, що склеюються):

$$(1-2) \rightarrow (\overline{x_1} \overline{x_2})_3 + (\overline{x_1} \overline{x_2} x_3)_{2,3} \quad (5.3)$$

$$(1-3) \rightarrow \text{не склеюються, бо не містять спільних ознак}$$

$$(1-4) \rightarrow \text{не склеюються}$$

$$(2-4) \rightarrow (\overline{x_1} x_3)_3 + (\overline{x_1} \overline{x_2} x_3)_{2,3} + (\overline{x_1} x_2 x_3)_{1,3} \quad (5.4)$$

$$(2-6) \rightarrow (\overline{x_2} x_3)_2 + (\overline{x_1} \overline{x_2} x_3)_{2,3} \quad (5.5)$$

$$(3-4) \rightarrow (\overline{x_1} x_2)_1 + (\overline{x_1} x_2 x_3)_{1,3} \quad (5.6)$$

$$(3-7) \rightarrow (x_2 \overline{x_3})_1 + (x_1 x_2 \overline{x_3})_{1,3} \quad (5.7)$$

$$(4-8) \rightarrow (x_2 x_3)_{1,3} + (x_1 x_2 x_3)_{1,2,3} \quad (5.8)$$

$$(5-6) \rightarrow (x_1 \overline{x_2})_2 \quad (5.9)$$

$$(6-8) \rightarrow (x_1 x_3)_2 + (x_1 x_2 x_3)_{1,2,3} \quad (5.10)$$

$$(7-8) \rightarrow (x_1 x_2)_{1,3} + (x_1 x_2 x_3)_{1,2,3} \quad (5.11)$$

Отже, ми виконали всі можливі склеювання мінтермів. Інші пари мінтермів не склеюються. На основі результатів склеювання (5.3) - (5.11) записуємо новий варіант допоміжної функції **Z**. Для цього спочатку записуємо диз'юнкцію імплікант, отриманих при склеюванні (в даному випадку імпліканти - це кон'юнкції двох змінних); далі дописуємо диз'юнкцію мінтермів, що фігурують в результатах склеювання (5.3) - (5.11). **При цьому не допускаємо повторення однакових (включно з ознаками) імплікант і мінтермів.** Далі до отриманого результату необхідно обов'язково дописати диз'юнкцію тих мінтермів, які взагалі не були задіяні при склеюванні (аналіз на наявність таких мінтермів проводиться обов'язково!!!) - в нашому прикладі таких мінтермів немає.

Діючи в описаній послідовності, для нашого прикладу отримаємо (як і в (2), всі кон'юнкції в (12) нумеруємо:

$$Z = (\overline{x_1} \overline{x_2})_3 + (\overline{x_1} x_3)_3 + (\overline{x_2} x_3)_2 + (\overline{x_1} x_2)_1 + (x_2 \overline{x_3})_1 + (x_2 x_3)_{1,3} + (x_1 \overline{x_2})_2 + \\ + (x_1 x_3)_2 + (x_1 x_2)_{1,3} + (\overline{x_1} \overline{x_2} x_3)_{2,3} + (\overline{x_1} x_2 x_3)_{1,3} + (x_1 x_2 \overline{x_3})_{1,3} + (x_1 x_2 x_3)_{1,2,3} \quad (5.12)$$

Наступним кроком є процедура *поглинання*.

2) поглинання - кожен мінтерм з (12) перевіряємо на предмет його поглинання одною з новоутворених імплікант. Причому поглинання можливе тільки за умови повного збігу ознак імпліканти і мінтерма. Поглинуті мінтерми викреслюємо з допоміжної функції **Z**. Так, в нашому прикладі мінтерм 11 поглинається імплікантою 6, а мінтерм 12 - імплікантою 9 (див. (12)). Викресливши згадані мінтерми з (12) і перенумерувавши заново кон'юнкції, отримаємо остаточно на першому етапі пошуку простих імплікант:

$$Z = (\overline{x_1} \overline{x_2})_3 + (\overline{x_1} x_3)_3 + (\overline{x_2} x_3)_2 + (\overline{x_1} x_2)_1 + (x_2 \overline{x_3})_1 + (x_2 x_3)_{1,3} + (x_1 \overline{x_2})_2 + \\ + (x_1 x_3)_2 + (x_1 x_2)_{1,3} + (\overline{x_1} \overline{x_2} x_3)_{2,3} + (x_1 x_2 x_3)_{1,2,3} \quad (5.13)$$

2-ий етап: - на цьому етапі здійснюємо склеювання імплікант, отриманих при виконанні попереднього етапу (в даному прикладі це кон'юнкції двох змінних). При цьому діємо аналогічно до 1-го етапу:

1) склеювання - здійснюємо на основі (5.13) по відношенню до кон'юнкцій двох змінних:

$$(4 - 9) \rightarrow (x_2)_1 + (x_1 x_2)_{1,3} \quad (5.14)$$

$$(5 - 6) \rightarrow (x_2)_1 + (x_2 x_3)_{1,3} \quad (5.15)$$

Інші пари кон'юнкцій двох змінних в (5.13) не склеюються. До нового варіанту допоміжної функції Z увійде: **1)** - новоутворена імпліканта $(x_2)_1$ (вона однакова (*повторяється*) в результатах склеювання (14), (15)); **2)** кон'юнкції 6, 9 з (5.13), які брали участь в склеюванні і не були відразу (за умови повного збігу ознак) поглинуті новоутвореними імплікантами; **3)** кон'юнкції з (5.13), що не брали участь в склеюванні на цьому етапі, а саме (див. (5.13)): 1, 2, 3, 7, 8, 10, 11.

Отже в нашому прикладі, після операції склеювання і перенумерування новоутворених кон'юнкцій, допоміжна функція Z набуде вигляду:

$$Z = (x_2)_1 + (\overline{x_1} \overline{x_2})_3 + (\overline{x_1} x_3)_3 + (\overline{x_2} x_3)_2 + (x_2 x_3)_{1,3} + (x_1 \overline{x_2})_2 + (x_1 x_3)_2 + (x_1 x_2)_{1,3} + (\overline{x_1} \overline{x_2} x_3)_{2,3} + (x_1 x_2 x_3)_{1,2,3} \quad (5.16)$$

2) поглинання - всі кон'юнкції в (5.16) перевіряємо на предмет їх поглинання *новоутвореними* імплікантами - в даному прикладі імплікантою $(x_2)_1$. Оскільки поглинання ми здійснюємо тільки при повному збігу ознак, приходимо до висновку, що жодна з кон'юнкцій виразу (5.16) новоутвореною імплікантою *не поглинається*. Тому, а також через те, що новоутворена імпліканта містить тільки одну змінну, вираз (5.16) є остаточним на етапі знаходження простих імплікант за Квайном. Тобто приходимо до висновку, що вираз (5.16) містить тільки прості імпліканти, які більше не склеюються між собою.

Контрольні запитання

1. Дайте інтерпретацію імпліканти і покриття Квайна.
2. Поясніть суть мінімізації логічних функцій методом Квайна.

Рекомендована література

1. William J. Dally, John W. Poulton Digital Systems Engineering. Cambridge University Press New York, NY, USA, 2008. ISBN:052106175X 9780521061759
2. Самофалов К.Г., Корнейчук В.И., Тарасенко В.П. Цифровые ЭВМ. Теория и проектирование. - К.: Вища шк., 1989.
2. Б.Є.Рицар. Цифрова техніка. - К.: НМК ВО, 1991.

Практичне заняття № 6

Тема: Факторизація.

Мета: Закріпити знання та навички факторизації логічних виразів.

Завдання

1. Факторизувати логічну функцію $y = \bar{x}_1 x_2 x_3 x_4 \bar{x}_5 + \bar{x}_3 x_4 + x_1 x_4 x_5 + x_1 \bar{x}_2 x_4$.

Коефіцієнт навантажуваності $k_H = 3$, коефіцієнт розгалуження $k_P = 4$.

Теоретичні відомості

Логічні мікросхеми мають обмежену навантажувальну здатність і кількість входів та виходів. Тому перед синтезом логічного пристрою, який описується логічною функцією, необхідно врахувати ці обмеження – факторизувати логічний вираз. Для факторизації виходів використовуються коефіцієнт навантажуваності k_H , входів – коефіцієнт розгалуження k_P .

Процес факторизації формалізується за допомогою таблиці імплікант $\varphi_1, \varphi_2, \dots, \varphi_n$, які покривають логічну функцію. Логічна функція при цьому представляється у нормальній диз'юнктивній формі.

Факторизація виходів логічної функції з n змінними полягає у складанні таблиці, яка складається з $2n$ стовпчиків – для змінних і їх інверсій, а також m рядків для імплікант. В побудовану таблицю, на перетині імпліканти і логічної змінної ставиться позначка, за умови, що змінна входить до даної імпліканти. При заданому коефіцієнті k_H факторизації підлягають джерела змінних, які мають більше, ніж k_H виходів. Визначення цих змінних полягає у підрахунку позначок в кожному стовпчику таблиці. Якщо сума позначок у стовпчику більша, ніж k_H , змінна підлягає факторизації. Ці стовпчики відповідають

додатковим вентилям, які повинні бути зв'язані з джерелом цієї змінної. Для врахування цих зв'язків додати до таблиці відповідну кількість рядків і стовпчиків і записати до них додаткові імпліканти для змінної, яка факторизується.

Факторизація входів при заданому коефіцієнті розгалуження k_p здійснюється аналогічно розподіленням надлишкових одиниць по додатковим рядкам.

Приклад.

Факторизувати

логічну

функцію

$y = \bar{x}_1 x_2 x_3 x_4 \bar{x}_5 + \bar{x}_3 x_4 + x_1 x_4 x_5 + x_1 \bar{x}_2 x_4$. Коефіцієнт навантажуваності $k_H = 3$, коефіцієнт розгалуження $k_p = 4$.

| | x_1 | x_2 | x_3 | x_4 | x_5 | \bar{x}_1 | \bar{x}_2 | \bar{x}_3 | \bar{x}_4 | \bar{x}_5 |
|-------------|-------|-------|-------|-------|-------|-------------|-------------|-------------|-------------|-------------|
| φ_1 | | + | + | + | | + | | | | + |
| φ_2 | | | | + | | | | + | | |
| φ_3 | + | | | + | + | | | | | |
| φ_4 | + | | | + | | | + | | | |

| | x_1 | x_2 | x_3 | x_4 | x_5 | \bar{x}_1 | \bar{x}_2 | \bar{x}_3 | \bar{x}_4 | \bar{x}_5 | Ψ_1 |
|-------------|-------|-------|-------|-------|-------|-------------|-------------|-------------|-------------|-------------|----------|
| φ_1 | | + | + | | | + | | | | + | + |
| φ_2 | | | | | | | | + | | | + |
| φ_3 | + | | | | + | | | | | | + |
| φ_4 | + | | | + | | | + | | | | |
| Ψ_1 | | | | + | | | | | | | |

| | x_1 | x_2 | x_3 | x_4 | x_5 | \bar{x}_1 | \bar{x}_2 | \bar{x}_3 | \bar{x}_4 | \bar{x}_5 | Ψ_1 | Ψ_2 |
|-------------|-------|-------|-------|-------|-------|-------------|-------------|-------------|-------------|-------------|----------|----------|
| φ_1 | | | | | | | | | | | + | + |
| φ_2 | | | | | | | | + | | | + | |
| φ_3 | + | | | | + | | | | | | + | |
| φ_4 | + | | | + | | | + | | | | | |
| Ψ_1 | | | | + | | | | | | | | |
| Ψ_2 | | + | + | | | + | | | | + | | |

Контрольні запитання

1. Дати визначення факторизації.
2. Дати визначення коефіцієнту навантаженості та коефіцієнту розгалуження.

Рекомендована література

1. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретна математика для інженера. – М.: Энергоатомиздат, 1988.
2. Гаврилов С.П. Сапоженко А.А. Збірник задач по дискретній математиці. – М.: Наука, 1978.
3. Нефедов В.Н., Осипова В.А. Курс по дискретній математиці. – М.: Издательство МАИ, 1992.
4. Кук Д., Бейз Г. Компьютерная математика. – М.: Наука, 1990.

Практичне заняття № 7

Тема: Перетворювачі кодів (дешифратори).

Мета: Вивчення принципів проектування дешифраторів в заданому базисі логічних елементів, а також дослідження функціонування спроектованих дешифраторів та їх інтегральних схем.

Завдання

1. Розробити логічну схему перетворювача двійково-десятичного коду у семирозрядний код керування семисегментними цифровими індикаторами.
2. Розробити схему перетворювача двійкового чотирьохрозрядного коду в код Грея та коду Грея в двійковий код.
3. Розробити перетворювач двійкового чотирьохрозрядного коду в код 2 з 5 та коду 2 з 5 в двійковий код.
4. Розробити перетворювач двійкового чотирьохрозрядного коду в додатковий код.
5. Використовуючи дешифратор K155ИД3, призначений для перетворення чотирьохрозрядного двійкового коду в напругу низького рівня, що з'являється на одному з шістнадцяти виходів $0 \div 15$, реалізувати логічну функцію $y = x_1 \bar{x}_2 \bar{x}_3 + x_1 x_3 + \bar{x}_1 x_2$. Для реалізації функції використовувати багатовходові логічні елементи з множини І, І-НІ, АБО, АБО НІ.
6. Використовуючи 4-розрядний дешифратор K155ИД3, розробити принципову схему дешифратора для перетворення п'ятирозрядного коду $x_1 - x_5$ з напругою низького рівня на одному з 32 виходів.
7. Використовуючи мікросхему K155ИД4, розробити: а) принципову схему демультиплексора $1 \rightarrow 8$; б) дешифратора 3×8 .

8. Використовуючи дешифратори серії 564ИД1, розробити принципову схему дешифрації шестирозрядного двійкового коду на 64 виходи.

9. Використовуючи пріоритетний шифратор 10×4 (555ИВ3) та перетворювач двійкового коду в код семисегментних індикаторів (К514ИД1), розробити принципову схему, що дозволяє на семисегментному індикаторі відобразити десяткову цифру при натисненні відповідної клавіші.

10. Комбінаційна схема визначена за допомогою наступних рівнянь: $y_1 = x_1x_2 + \bar{x}_1\bar{x}_2\bar{x}_3$; $y_2 = x_1 + x_2 + \bar{x}_3$; $y_3 = \bar{x}_1x_2 + x_1\bar{x}_2x_3$. Розробити схему, що реалізує ці три рівняння за допомогою дешифратора та зовнішніх логічних елементів.

11. Задачу №2 розв'язати з використанням дешифратора та додаткових зовнішніх елементів.

12. Комбінаційна схема визначена наступними рівняннями: $y_1 = x_1x_2x_3 + \bar{x}_1\bar{x}_2x_3$; $y_2 = \bar{x}_1 + \bar{x}_2 + x_3 + x_4$; $y_3 = x_1 + x_2 + x_2x_3 + \bar{x}_1x_4$; $y_4 = x_1x_3x_4 + x_1\bar{x}_3\bar{x}_4 + x_2\bar{x}_3x_4 + x_2x_3\bar{x}_4$. Побудувати схему, що реалізує ці чотири рівняння за допомогою дешифратора та зовнішніх логічних пристроїв.

13. Розробити дешифратор на три входи та вісім виходів, використовуючи тільки логічні елементи АБО-НІ, та зобразити його логічну структуру.

14. Розробити схему шифратора, призначеного для перетворення десяткового коду, що знімається з клавіатури, у чотирьохрозрядний двійковий код.

Теоретичні відомості

Шифрування полягає у перетворенні m -розрядного коду з k_m байдужих наборів вхідних змінних в однозначно відповідний йому n -розрядний код з меншим числом розрядів ($n < m$) і байдужих наборів. Під байдужими наборами в будь-якому коді маються на увазі ті набори, які не використовуються при перетворенні. Пристрої, які виконують вказані операції, називаються відповідно *шифраторами* та *дешифраторами* (кодерами та декодерами).

Дешифратори і шифратори (як і елементи І, АБО, НЕ, І-НЕ, АБО-НЕ) є комбінаційними елементами: сигнали на їх виходах залежать від стану входів, такі елементи не зберігають попередній стан після зміни сигналу на входах, тобто не мають пам'ять. Дешифратори можуть бути повними й неповними. Повні дешифратори реагують на всі вхідні коди, неповні - на коди, величина яких не перевершує деякого заздалегідь встановленого значення. Виходи дешифраторів можуть бути прямими і інверсними.

Дешифратор призначений для перетворення двійкового коду на вході в керуючий сигнал на одному з виходів. Дешифратор широко застосовують в пристроях управління, в системах цифрової індикації з газорозрядними індикаторами, для побудови розподільників імпульсу.

Якщо дешифратор має n входів, то вихідних шин повинно бути $N=2^n$, так як для кожного із 2^n значень вхідного коду повинен відповідати одиничний сигнал на одному із виходів дешифратора. Дешифратор працює відповідно до таблиці 6.

Таблиця 6

| X1 | X2 | X3 | Y0 | Y1 | Y2 | Y3 | Y4 | Y5 | Y6 | Y7 |
|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Основу структури дешифратора можуть становити елементи І; вихід кожного з них є виходом дешифратора. Якщо цей вихід повинен бути збуджений, то на входах елемента І повинні збиратися логічні одиниці. При

цьому, розряди вхідного коду, в якому присутні логічні одиниці, повинні надходити на входи елемента І безпосередньо, а нульові розряди повинні інвертуватися.

Викладений принцип покладений в основу побудови схеми, зображеної на рис. 5. Логічна 1 на виході Y_0 повинна з'явитися, коли на входах X_3, X_2, X_1 присутній двійковий код 000 десятичного числа 0. Тому входи верхнього (за схемою) кон'юнктора повинні бути з'єднані з лініями $\overline{X_1}, \overline{X_2}, \overline{X_3}$, на кожній з яких є логічна 1, коли на входах $X_3=X_2=X_1=0$. Логічна 1, наприклад, на виході Y_2 повинна з'явитися, коли на входах X_3, X_2, X_1 встановлюється код 010 десятичного числа 2, тому входи відповідного кон'юнктора повинні бути з'єднані з лініями $\overline{X_1}, X_2, \overline{X_3}$, на кожній з яких є логічна 1, коли $X_3=0, X_2=1, X_1=0$. Аналогічно з'єднуються з лініями входи інших кон'юнкторів.

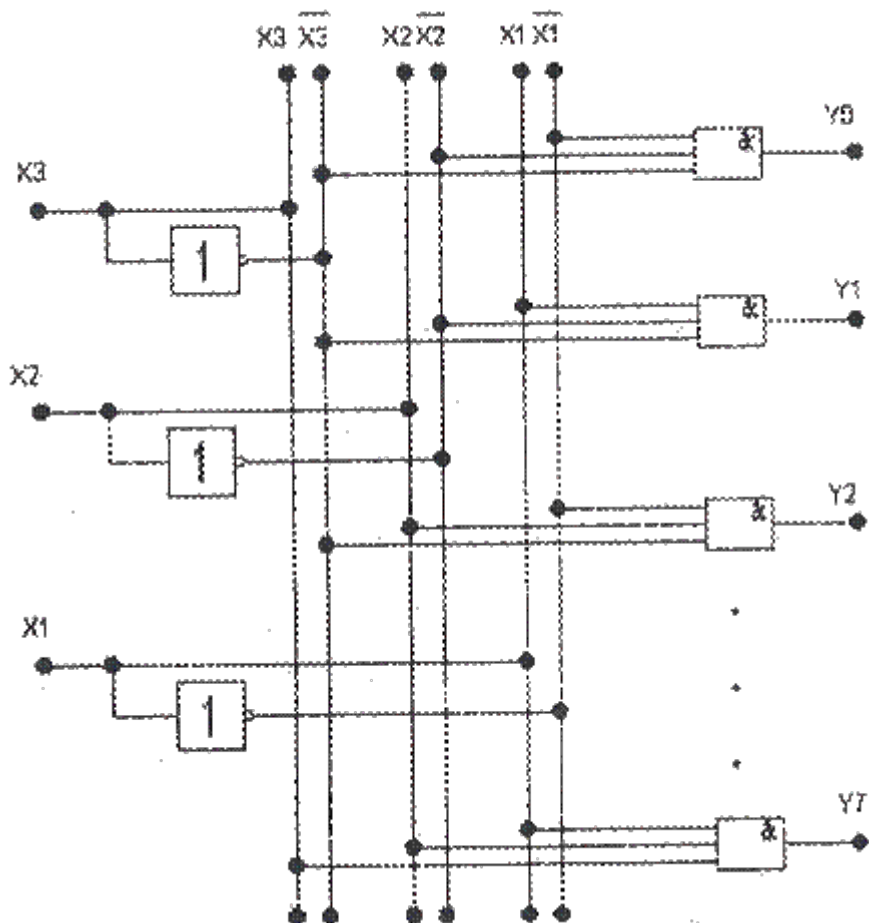


Рис. 5

Деякі типи дешифраторів мають інверсні виходи: на збудженому (активізованому) виході присутній логічний 0, в той час як на всіх інших - логічна 1. Такі дешифратори зручно використовувати, коли активним сигналом для вибору (запровадження в дію, ініціалізації) пристрою з виходу дешифратора є логічний 0.

Загальний випадок розширення розрядності дешифраторів ілюструє рис. 2. Лівий (за схемою) дешифратор постійно активізований логічною 1 на вході V. Кодами на його адресних входах може бути активізований (обраний) кожен з дешифраторів DC0...DC15. Вибір одного з виходів 0...15 кожного з них визначається кодом на об'єднаних входах 1, 2, 4, 8. Таким чином, кожен з 256 (28) виходів може бути активізований восьмирозрядним кодом, чотири розряди якого вибирають номер дешифратора, а чотири - номер його виходу.

Основне призначення дешифратора полягає в тому, щоб вибрати (адресувати, ініціалізувати) один об'єкт із безлічі, які знаходяться. Кожному об'єкту присвоюють конкретну адресу (номер). Коли на входи дешифратора надходить двійковий код адреси, відповідний елемент активується за рахунок появи логічного 0 на зв'язаному з ним виході дешифратора, а інші елементи залишаються заблокованими.

Можна передбачити, щоб з одного з виходів дешифратора на визначений блок надходив керуючий сигнал, коли на входах дешифратора з'являється визначений код, що відповідає, наприклад, перевищенню якогось параметра (температури, напруги і т.д.), що повинен бути приведений до нормального рівня зазначеним блоком.

Дешифратори широко використовуються для реалізації логічних функцій. Наприклад, для реалізації функції: $y = \vee 1, 3, 4, 7$ достатньо об'єднати виходи y_1, y_3, y_4, y_7 за допомогою чотирьохвходового елемента **АБО**.

Якщо виходи $y_0 \div y_{n-1}$ дешифратора інвертовані, то об'єднання виходів відповідних диз'юнкцій забезпечується елементами, що реалізують операцію **I-НІ**.

Використання дешифраторів як генераторів мінтермів корисно в тих випадках, коли необхідно реалізовувати систему логічних функцій. При використанні дешифраторів для реалізації логічних функцій відпадає необхідність їх мінімізації. Більш того, їх необхідно приводити до досконалої диз'юнктивної форми.

Контрольні запитання

1. Дайте визначення дешифратора та шифратора.
2. Як працює дешифратор?
3. Яку мінімальну розрядність повинен мати дешифратор для адресації 11-ти пристроїв?
4. Скільки входів повинен мати неповний дешифратор, що має 10 виходів?
5. В яких пристроях використовується дешифратор?
6. Як працює шифратор? При розв'язанні яких задач він використовується?
7. Який номер збудженого входу шифратора, якщо на виході встановився код 0110?
8. Як на базі дешифратора можуть бути реалізовані логічні функції?

Рекомендована література

1. Угрюмов Е. П. Цифровая схемотехника. – СПб.: БХВ-Петербург, 2002. – 46 с.
2. Шило В. Л. Популярныe микросхемы ТТЛ. М., Аргус, 1993.

Практичне заняття № 8

Тема: Мультиплексори – демультиплексори.

Мета: Отримання практичних навиків в побудові і контролі працездатності таких вузлів як мультиплексори, демультиплексори, перетворювачі коду.

Завдання

1. На основі мікросхеми K155КП2, що містить два незалежних мультиплексори $8 \rightarrow 1$.

2. Використовуючи мікросхему K155КП2 розробити логічні пристрої, що реалізують наступні функції: $y_1 = x_1x_2 + \bar{x}_1\bar{x}_2 = x_1 \otimes x_2$; $y_2 = x_1 \oplus x_2$.

3. Використовуючи мультиплексор K155КП7, розробити логічний пристрій, що реалізує функцію $y = x_1\bar{x}_2x_3 + x_2\bar{x}_3 + \bar{x}_1x_2$.

4. Реалізувати за допомогою мультиплексорів $4 \rightarrow 1$ логічну функцію: $y = x_1x_3\bar{x}_4x_5 + \bar{x}_1\bar{x}_4x_5 + \bar{x}_1\bar{x}_3x_4 + x_3\bar{x}_4 + \bar{x}_2x_3$.

5. На базі мультиплексорів $8 \rightarrow 1$ реалізувати схему мультиплексора $64 \rightarrow 1$.

6. Реалізувати на мультиплексорах $4 \rightarrow 1$ та демультиплексорах $1 \rightarrow 4$ комутатор, що виконує з'єднання будь-якого з восьми джерел інформації з будь-яким з 24 приймачів інформації.

7. Реалізувати логічну функцію $y = x_1x_2\bar{x}_3 + \bar{x}_1x_3$ на мультиплексорах $2 \rightarrow 1$ та $4 \rightarrow 1$. Порівняти отримані результати.

8. Використовуючи мікросхему 564КП2, розробити схеми:

1) мультиплексора $8 \rightarrow 1$ та демультиплексора $1 \rightarrow 8$;

2) мультиплексора $4 \rightarrow 1$ та демультиплексора $1 \rightarrow 4$.

9. Організувати роботу мультиплексора, на виході якого формується тактова послідовність імпульсів – чотири нулі – чотири одиниці.

10. Сформувати послідовність адресних сигналів, що дозволяють знімати інформацію з одnobайтової шини даних у послідовності: спочатку чотири непарних біта, потім – чотири парних.

11. У суднових системах сигналізації є пристрої («Туман», «Пульсар-МТ»), що призначені для формування світлових та звукових сигналів при виконанні судном маневрів у тумані. На рис. 6 приведені часові діаграми декількох варіантів сигналів, що формуються, з вказівкою їх часових інтервалів. Розробити алгоритм формування адресації мультиплексора та схему його включення для реалізації кожної з вказаних часових послідовностей імпульсів.

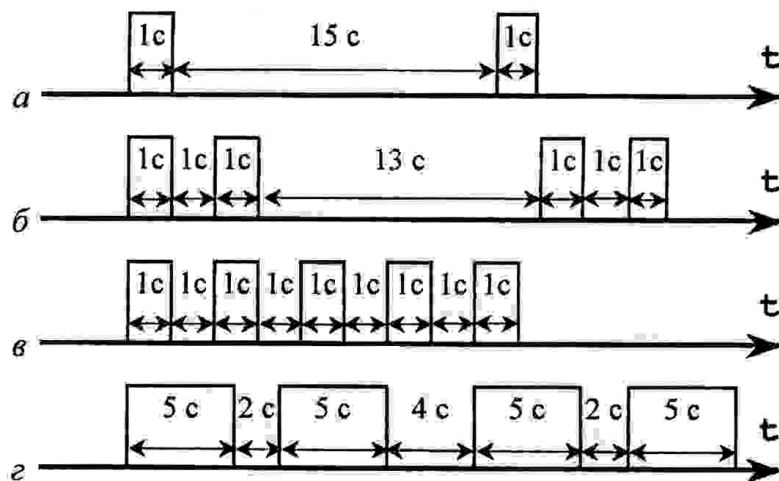


Рис. 6

12. Реалізувати функцію трьох змінних $y = \bar{x}_1\bar{x}_2x_3 + \bar{x}_1x_2x_3 + \bar{x}_1x_2\bar{x}_3 + x_1x_2x_3$ за допомогою мультиплексора, що має чотири інформаційні входи.

13. Реалізувати функцію чотирьох змінних $y = f(x_1, x_2, x_3, x_4) = \sum 0, 1, 5, 6, 7, 9, 10, 14, 15$ за допомогою мультиплексора, що має чотири інформаційні входи.

14. Реалізувати за допомогою мультиплексорів $4 \rightarrow 1$ наступні булеві функції трьох змінних: $y_1 = f(x_1, x_2, x_3) = \sum 0, 2, 3, 5, 7$, $y_2 = f(x_1, x_2, x_3) = \sum 1, 3, 4, 6, 7$, $y_3 = f(x_1, x_2, x_3) = \sum 0, 2, 4, 5, 6, 7$. Визначити

комбінаційні керуючих адресних змінних, при яких додаткова логіка була б мінімально.

15. Реалізувати за допомогою чотирьохвходових мультиплексорів та логічних елементів І-НІ наступні булеві функції чотирьох змінних:

$$y_1 = f(x_1, x_2, x_3, x_4) = \sum 0, 1, 3, 5, 6, 8, 9, 11, 12, 13 \text{ керуючі змінні } x_1 \text{ та } x_2;$$

$$y_2 = f(x_1, x_2, x_3, x_4) = \sum 0, 7, 8, 9, 10, 11, 13, 14, 15 \text{ керуючі змінні } x_2 \text{ та } x_3;$$

$$y_3 = f(x_1, x_2, x_3, x_4) = \sum 0, 1, 3, 5, 9, 10, 11, 13, 14, 15 \text{ керуючі змінні } x_3 \text{ та } x_4;$$

$$y_4 = f(x_1, x_2, x_3, x_4) = \sum 1, 8, 9, 10, 12, 13, 14, 15 \text{ керуючі змінні } x_1 \text{ та } x_4.$$

Теоретичні відомості

Мультиплексор – це цифровий комбінаційний пристрій, який забезпечує перетворення інформації, що задана на входах $d_7 \dots d_0$ у паралельному форматі, в послідовний формат на виході Y . Умовне позначення мультиплексора в електронних схемах приводиться на рис. 7. Входи A називають адресними (адресна шина), а входи D – входами даних (шина даних). Допоміжний вхід V називають керуючим або дозволяючим. З його урахуванням робота мультиплексора, приведенного на рис. 7, описуватиметься наступною логічною функцією:

$$Y = \overline{v} \cdot (\overline{a_1} \overline{a_0} d_0 + \overline{a_1} a_0 d_1 + a_1 \overline{a_0} d_2 + a_1 a_0 d_3) . \quad (8.1)$$

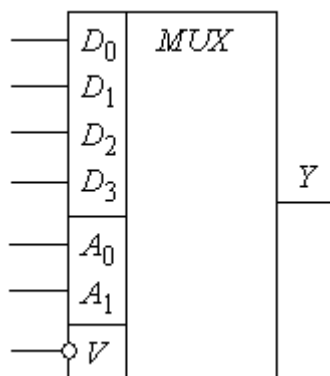


Рис. 7

Дозволяючий вхід V розширює можливості пристрою. Він дає можливість синхронізувати роботу мультиплексора з іншими схемами, а також використовується для нарощування розрядності адресних сигналів. Логічна схема мультиплексора, реалізована у базисі елементів **I-АБО-НІ**, приведена на рис. 8.

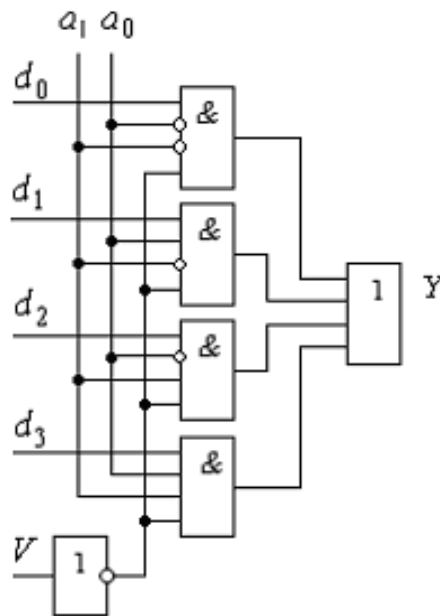


Рис. 8

Оскільки мультиплексор є пристроєм логіки, то на його базі можуть реалізовуватись різні логічні функції. Як приклад, скористаємось формулою (8.1). Припустимо, що необхідно реалізувати логічну функцію:

$$y = x_0 \cdot x_1 + \overline{x_0} \cdot \overline{x_1}. \quad (8.2)$$

Порівнюючи формули (8.1) і (8.2), бачимо, що, присвоївши адресним сигналам a_0 , a_1 значення x_0 , x_1 відповідно і забезпечивши виконання умов $d_1 = d_2 = 0$, $d_0 = d_3 = 1$, $v = 0$, одержимо схему пристрою (рис. 3.10), який реалізує функцію (8.2).

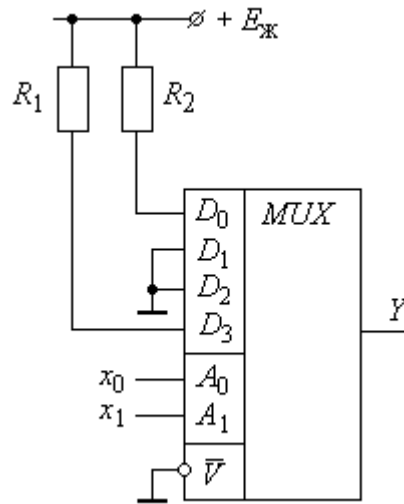


Рис. 9

Нескладні перетворення логічних функцій дають можливість реалізовувати на мультиплексорах функції з кількістю змінних, що перевершує кількість адресних входів. Виконується наступна послідовність перетворень:

- у мінімізованій диз'юнктивній нормальній формі функції виділяються змінні, які мають найвищий ранг, тобто повторюються в найбільшій кількості диз'юнкцій. Наприклад, вибирається m змінних;
- виконується перетворення функції так, щоб забезпечити присутність виділених змінних в усіх диз'юнкціях. Для цього кожна диз'юнкція домножається на $1 = x + \bar{x}$;
- проводиться перегрупування змінних логічної функції таким чином, щоб виділені змінні були винесені за дужки у відповідних групах диз'юнкцій.

У результаті виконаних перетворень одержується формула вихідної функції, яка реалізується на мультиплексорі “з $M = 2^m$ в 1”, якщо на його адресні входи подати виділені змінні, а на інформаційних входах забезпечується виконання тих логічних виразів, які залишились в дужках. Якщо вирази в дужках є функціями декількох змінних, то їх реалізація може бути забезпечена або за допомогою іншого мультиплексора, або на основі типових логічних елементів.

У більшості серій мікросхем мультиплексори, що виготовляються у вигляді самостійних виробів, мають кількість інформаційних входів, не

перевищуючу шістнадцяти. Збільшення їх кількості досягається різними прийомами з'єднання окремих мультиплексорів. Наприклад, на рис. 10 приведена схема паралельного нарощування.

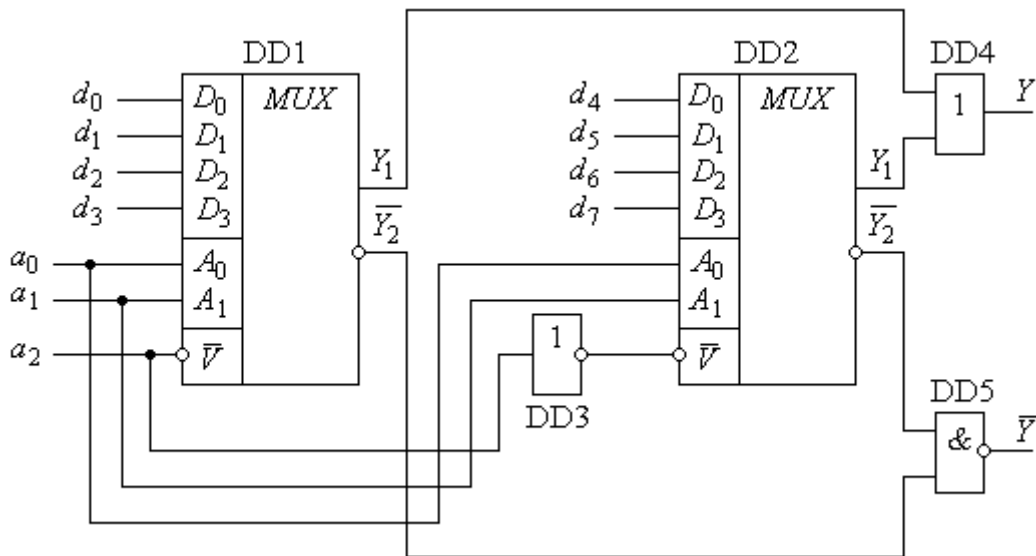


Рис. 10

Для збільшення кількості інформаційних входів необхідно підвищувати кількість адресних провідників (ємність адресного простору, адресної шини). У схемі, приведений на рис. 10, розширення адресного простору досягається використанням входу V , який, об'єднаний в обох мультиплексорах через інвертор DD3, виступає старшим розрядом адресної шини. При $a_2 = 0$ зміна сигналів на a_0, a_1 дозволить вибрати входи $d_0 \dots d_3$ шини даних і через елемент **АБО** (DD4) передавати на вихід Y . Мультиплексор DD2 при цьому відключений високим рівнем сигналу на виході DD3. При $a_2 = 1$ зміною a_0, a_1 забезпечується комутація входів $d_4 \dots d_7$ на вихід Y . При наявності в мультиплексорах інверсних виходів їх об'єднання забезпечується елементом **І-НІ** (DD5).

При необхідності суттєвого нарощування кількості комутованих інформаційних входів використовується пірамідальний спосіб нарощування. Приклад його реалізації приведений на рис. 11. Ця схема в реалізації є більш гнучкою. Адресні входи мультиплексорів нижнього рівня DD1...DD4 з'єднуються паралельно і керуються молодшими розрядами адресного

простору. Їх кількість визначається тільки кількістю інформаційних входів кожного мультиплексора. Мультиплексор верхнього ступеня задає старші розряди адресного простору, завдяки яким він забезпечує комутацію виходу кожного з мультиплексорів $Y_0 \dots Y_3$ на загальний вихід Y .

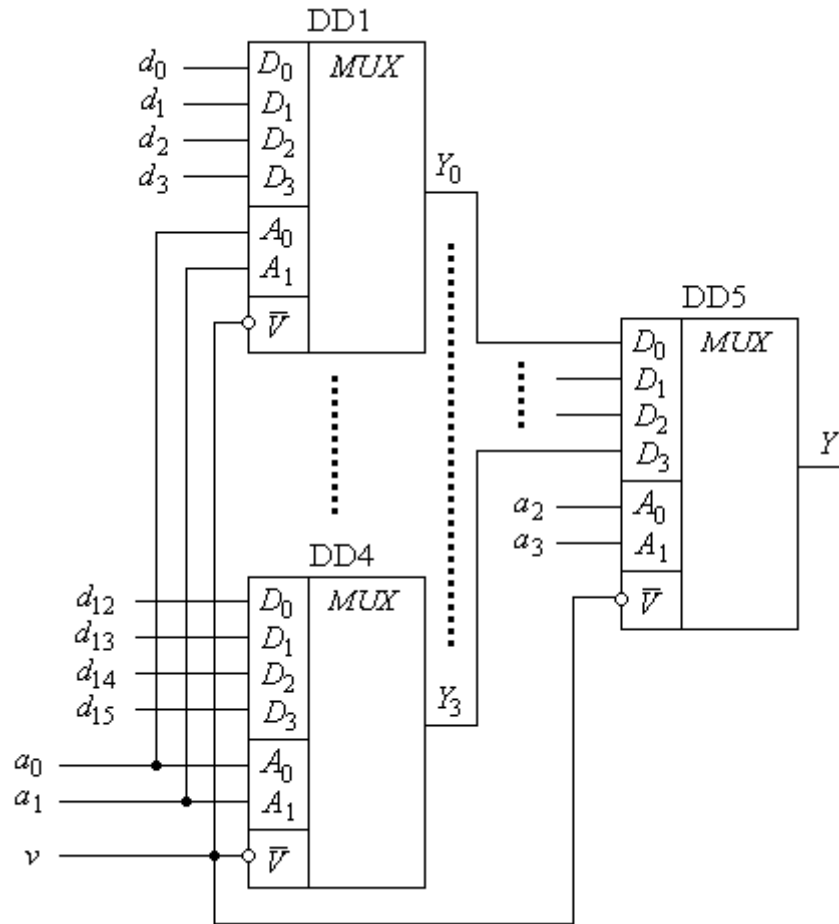


Рис. 11

Контрольні запитання

1. Які вузли називаються вузлами комутації і контролю?
2. Які функції в цифрових пристроях виконує дешифратор?
3. Чи можна використовувати мультиплексори для реалізації логічних функцій?
4. Як здійснюється керування роботою мультиплексора?

5. За допомогою якого вузла можна перетворити десятковий код в двійково-десятковий?

6. Скільки виходів може мати демультіплексор, якщо кількість адресних входів дорівнює 4?

Рекомендована література

1. Системне програмування. Системні сервісні компоненти: навч. посібник / О.С. Дерев'янка, С.Г. Межерицький, С.Ю. Гавриленко, А.М. Клименко. – Харків: НТУ «ХП», 2009. –160 с.

2. Шеховцев В.А. Операційні системи. – К.: ВНУ, 2005. – 576 с.

3. Абель П. Язык ассемблера для IBM PC и программирование. – М.: Высш. шк., 2008.– 447 с.

Практичне заняття № 9

Тема: Суматори – компаратори.

Мета: Вивчення комбінаційних мікросхем: суматорів і компараторів кодів; їх алгоритмів роботи; типових схем включення.

Завдання

1. Розглянути мікросхему K155ИМ3. Пояснити, як використовувати вказаний чотирьохрозрядний суматор для додавання:

- а) двох чотирьохрозрядних чисел;
- б) двох дворозрядних та двох однорозрядних чисел;
- в) двох восьмирозрядних (байтових) чисел.

2. Використовуючи мікросхему K155ИМ3. Розробити схему пристрою, що дозволяє здійснювати як додавання, так і віднімання напівбайтових слів.

3. Спроектувати комбінаційну схему пристрою, за допомогою якого можна здвійснювати доповнення до 9 кожної десяткової цифри.

4. Розробити перетворювач чотирьохрозрядного двійкового коду 8-4-2-1 у код «з надлишком 3».

5. Розробити пристрій двійкового множення на суматорах та логічних елементах І, що дозволяє множити чотирьохрозрядне число A на трьохрозрядне число B .

6. Розробити схему пристрою, що дозволяє отримувати квадрати будь-якого трьохрозрядного двійкового числа.

7. Використовуючи мультиплексор K155КП2, розробити принципові схеми дворозрядного суматора та дворозрядного від'ємника.

8. Використовуючи логічні елементи *ВИКЛЮЧАЮЧЕ АБО* та елементи розробити схему дворозрядного компаратора.

9. Розробити принципову схему суматора, який здійснює додавання однорозрядних десяткових чисел.

10. Використовуючи результат попередньої задачі, розробити схему пристрою для додавання двох дворозрядних десяткових чисел.

11. Використовуючи чотирьохрозрядний суматор та додаткові логічні елементи, розробити схему чотирьохрозрядного компаратора двійкових чисел.

12. З використанням чотирьохрозрядного суматора К155ИМ3, розробити схему пристрою додавання та віднімання двох дванадцятирозрядних двійкових чисел.

13. Використовуючи чотирьохрозрядні компаратори К155ИП2, розробити схему пристрою порівняння двох дванадцятирозрядних двійкових чисел A та B . Ці схеми повинні показувати, чи є: $A > B$, $A = B$, $A < B$. Розширити схеми таким чином, щоб можна було порівнювати чотирьохрозрядні числа.

14. Розробити комбінаційну логічну схему, яка дозволить здійснювати перетворення чотирьохрозрядних двійкових чисел у додатковий код.

Теоретичні відомості

Найпростішою схемою є *напівсуматор*, який виконує операцію знаходження суми двох однорозрядних слів. Роботу напівсуматора описує Табл. 7, з якої знаходимо логічні рівняння для суми і для переносу:

$$S = a_0 \cdot \overline{b_0} + \overline{a_0} \cdot b_0 ;$$
$$P = a_0 \cdot b_0 .$$

Таблиця 7

| a_0 | b_0 | S | P |
|-------|-------|-----|-----|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

При знаходженні суми багаторозрядних слів напівсуматор може використовуватись лише в молодшому розряді. Для знаходження суми старших розрядів схема повинна мати три входи – два інформаційні і третій – вхід переносу з молодших розрядів. Схема, що відповідає цим вимогам, називається *повним суматором*, а логіку її роботи описує Табл. 8.

Таблиця 8

| N | a_i | b_i | p_{i-1} | S_i | P_{i+1} |
|-----|-------|-------|-----------|-------|-----------|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 2 | 0 | 1 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 | 0 | 1 |
| 4 | 1 | 0 | 0 | 1 | 0 |
| 5 | 1 | 0 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 | 1 |

З Табл. 8 знаходимо наступні рівняння:

$$S_i = \vee 1, 2, 4, 7 = p_{i-1} (\overline{a_i} \cdot \overline{b_i} + a_i \cdot b_i) + \overline{p_{i-1}} (a_i \cdot \overline{b_i} + \overline{a_i} \cdot b_i) ;$$

$$P_i = \vee 3, 5, 6, 7 = p_{i-1} (\overline{a_i} \cdot b_i + a_i \cdot \overline{b_i} + a_i \cdot b_i) + \overline{p_{i-1}} a_i \cdot b_i = a_i \cdot b_i + p_{i-1} (a_i \oplus b_i)$$

для побудови логічної схеми однорозрядного повного суматора.

Знаходження суми багаторозрядних слів може виконуватись як послідовно, так і паралельно. Для послідовного виконання операції використовується один повний суматор, загальний для всіх розрядів. Для виконання операції суми слова подаються на інформаційні розряди послідовно, починаючи з молодших розрядів (рис. 12).

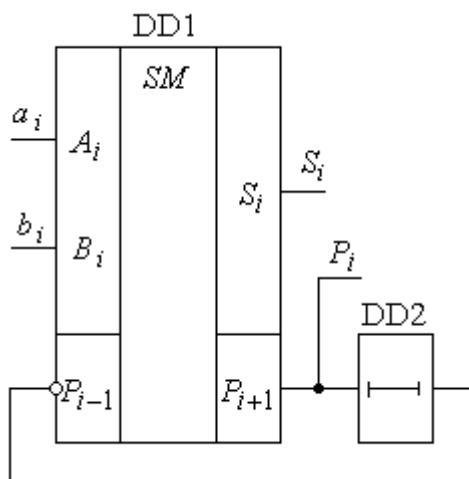


Рис. 12

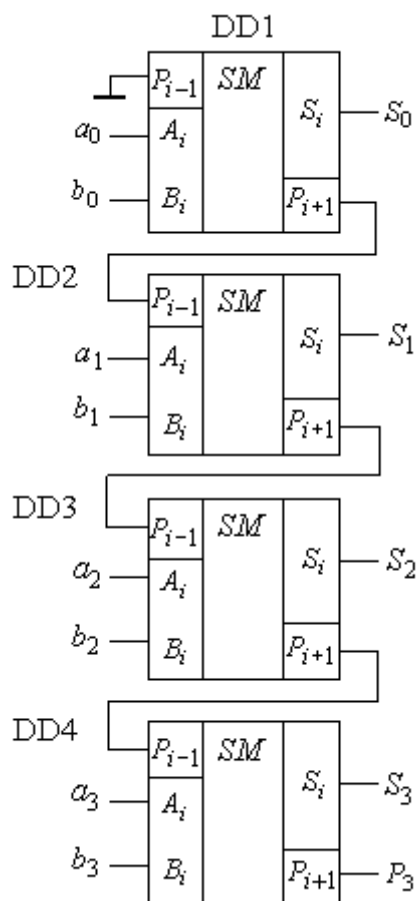


Рис. 13

Для врахування результату операції з попереднім розрядом вхід та вихід переносу об'єднується через елемент затримки DD2. Цей елемент повинен забезпечити затримку та зберігання сигналу P_{i+1} на один такт виконання операції. Синхронно з інтервалом в один такт повинні подаватись і вхідні дані a_i, b_i .

На рис. 13 приведена схема організації чотирьохрозрядного паралельного суматора з послідовним переносом. Для кожного розряду в ній використовуються окремі однорозрядні повні суматори, які з'єднані між собою послідовно по виходах-входах переносу. Незважаючи на паралельну одночасну подачу всіх розрядів слів, операція в i -му розряді починає виконуватись тільки після того, як поступить сигнал переносу з $(i-1)$ -го розряду. Звідси витікає, що швидкодія суматора визначатиметься сумою затримок передачі сигналу переносу з молодшого розряду на вихід суматора старшого розряду.

Нарощування розрядності суматорів виконується на схемі, приведений на рис. 13. Таблиця станів для суматорів за своєю сутністю є таблицею суми двох чисел (Див. табл. 7 – табл. 8). Вона має простий вигляд і зручна в користуванні для чисел невеликої розрядності. Суматор К1533ИМЗ, як і всі комбінаційні схеми, має широкі функціональні можливості. Перш за все, він може використовуватись для знаходження суми чисел з меншою розрядністю. Схема входів-виходів при цьому може бути різноманітною. Його можливо використовувати і для віднімання двох чисел. Ця операція виконується шляхом додавання зменшуваного і зменшуючого, представленого у доповнюючому коді. Для виконання цієї операції зменшуюче $b_3 b_2 b_1 b_0$ з допомогою інверторів перетворюється в зворотній код, а потім до нього по входу p_0 додається одиниця.

Операції знаходження суми та різниці можна поєднати в одному пристрої, як показано на рис. 14. Для виконання операції додавання на вхід P_{i-1} переносу подається сигнал низького рівня. Відповідно на входах елементів **ВИКЛ. АБО** число $b_3 b_2 b_1 b_0$ буде представлене в прямому коді. При подачі на вхід P_{i-1} сигналу високого рівня, як відмічалось раніше, число $b_3 b_2 b_1 b_0$ буде представлене у доповнюючому коді.

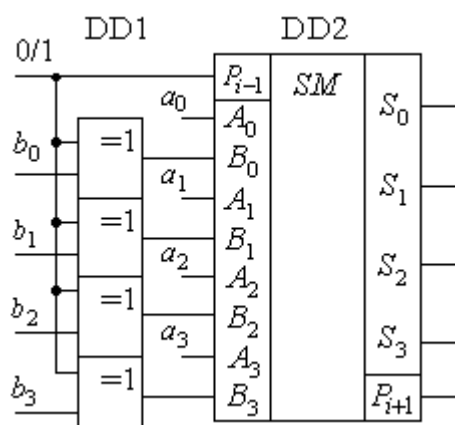


Рис. 14

При виконанні операції віднімання результат на виході суматора залежить від співвідношення між числами, що беруть участь в операції. При $a_3 a_2 a_1 a_0 > b_3 b_2 b_1 b_0$ результат одержується в прямому коді, а в розряді

переносу P_{i+1} з'являється **1**. Коли ж $a_3 a_2 a_1 a_0 < b_3 b_2 b_1 b_0$, результат отримується в зворотньому коді, а в розряді переносу P_{i+1} одержується **0**.

Компаратори – це пристрої, що виконують порівняння двох чисел.

Мікросхеми компараторів визначають не тільки рівність, але й нерівність двох чисел. Для цього мікросхема має три виходи: « $A > B$ », « $A < B$ » і « $A = B$ », на яких у залежності від співвідношення величин $A = a_3 a_2 a_1 a_0$, $B = b_3 b_2 b_1 b_0$ з'являється активний рівень сигналу.

Логічна функція, що реалізує алгоритм роботи компаратора для одного розряду:

$$y_i = a_i \cdot b_i + \overline{a_i} \cdot \overline{b_i} = \overline{a_i \oplus b_i}. \quad (9.1)$$

Для багаторозрядних чисел функція (9.1) повинна виконуватись для кожного розряду. Нерівність $A > B$ має місце тоді, коли в числі A одиниці дорівнює більш старший розряд, ніж у числі B . Для чотирьохрозрядного компаратора така нерівність може бути записана у вигляді:

$$(A > B) = a_3 \overline{b_3} + (\overline{a_3 \oplus b_3}) \cdot a_2 \overline{b_2} + (\overline{a_3 \oplus b_3}) \cdot (\overline{a_2 \oplus b_2}) \cdot a_1 \overline{b_1} + \\ + (\overline{a_3 \oplus b_3}) \cdot (\overline{a_2 \oplus b_2}) \cdot (\overline{a_1 \oplus b_1}) \cdot a_0 \overline{b_0}.$$

Побудова багаторозрядного компаратора можлива на базі суматора. Його реалізація можлива при виконанні операції віднімання.

Дійсно, в відповідності до законів арифметики при $A = B$ на виходах всіх розрядів суматора s_0, s_1, s_2, s_3 буде **0** при $P_{i+1} = 1$. При $A > B$ значення **1** буде як на виході переносу $P_{i+1} = 1$, так і хоча б на одному виході s_i . Тому ознакою $A > B$ може бути функція $y_> = p_{i+1} \cdot (\sum s_i)$. При $A < B$ результат наявності переносу $P_{i+1} = 0$.

Мікросхеми мають розширення по входу (входи « $A > B$ », « $A < B$ » і « $A = B$ »), що дає можливість як послідовного нарощування, так і паралельного. На рис. 15 приведена схема послідовного нарощування двох компараторів К561ИП2.

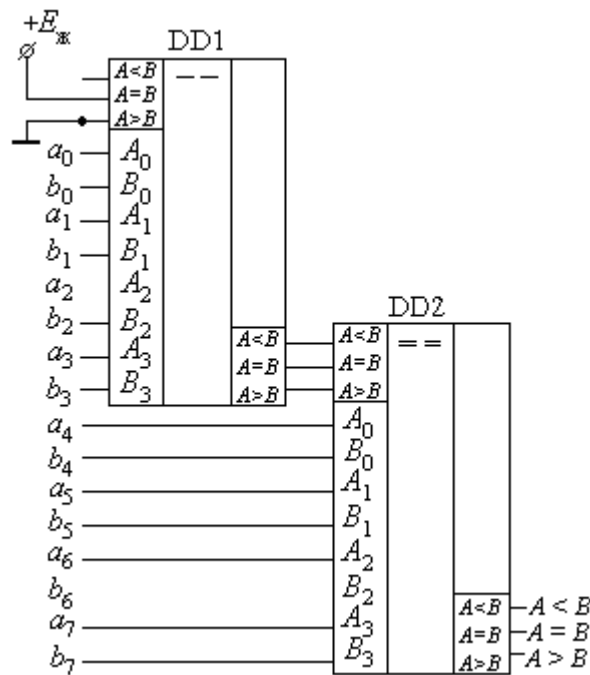


Рис. 15

Входи розширення першої мікросхеми необхідно з'єднати відповідно до рис. 15, а виходи з'єднуються безпосередньо з входами наступної мікросхеми. При такому способі нарощування компараторів затримки додаються, а стан виходів другого і послідуєчих компараторів залежить не тільки від побітних значень порівнюваних слів, а й від результату порівняння молодших розрядів цих слів. Тому значення виходів кожного з таких компараторів можуть бути описані рівняннями:

$$(A > B)_{\text{вих}} = (A > B)_{n-m} + (A = B)_{n-m} \cdot (A > B)_{\text{вх}} ;$$

$$(A = B)_{\text{вих}} = (A = B)_{n-m} \cdot (A = B)_{\text{вх}} ;$$

$$(A < B)_{\text{вих}} = (A < B)_{n-m} + (A = B)_{n-m} \cdot (A < B)_{\text{вх}} ;$$

де n, m – діапазон розрядів порівнюваних слів, які приєднані до компаратора.

При пірамідальному способі з'єднання тривалість затримок зменшується.

Компаратори широко використовуються в інформаційних системах для виділення необхідного слова в потоці цифрової інформації, для відмітки часу в часових пристроях, для виконання умовних переходів в обчислювальних пристроях. У пристроях автоматики компаратори використовуються для контролю виходу величин за межі допусків і т.п.

Контрольні запитання

1. Які відмінності суматора від напівсуматора?
2. Які функції виконує цифровий компаратор?
3. Які булеві функції використовуються при побудові компараторів?

Рекомендована література

1. Гусев В.Г., Гусев Ю.М. Электроника/М.: Высш. шк.,1991.-621с.
2. Каяцкас А.А. Основы радиоэлектроники/М.: Высш. шк.,1968.-646с.

Практичне заняття № 10

Тема: Тригери.

Мета: Вивчити принципи функціонування тригерів, отримати навички роботи з тригерами.

Завдання

1. Замалювати виходи RS -тригера на елементах **АБО-НІ** для вхідних сигналів, зображених на рис. 16. Прийняти наступні припущення: тривалість вихідних фронтів і спадів дорівнює нулю; середня затримка вентиля 10 нс; кожен інтервал часу також дорівнює 10 нс; початковий стан тригера – нульовий.

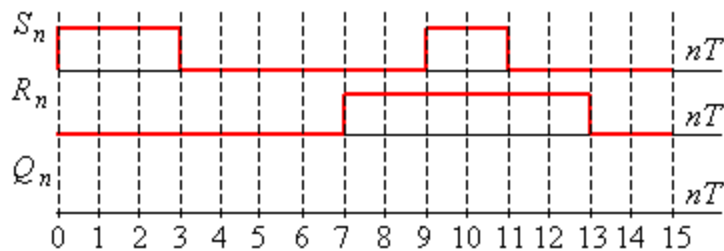


Рис. 16

2. Повторити вправу 1, використовуючи вхідні сигнали відповідно до рис. 17.

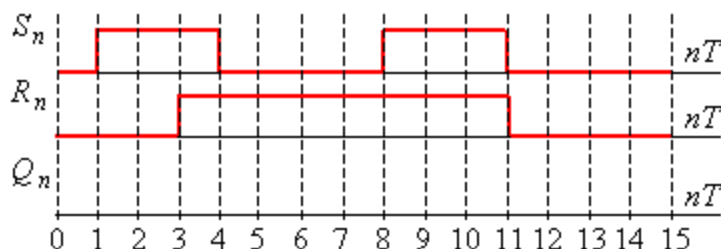


Рис. 17

3. Побудувати RS -тригер, використовуючи один елемент – D -тригер з позитивним фронтом типу КР1533ТМ2 (74ALS74А), без застосування інших компонентів.

4. Використовуючи елемент КР1533ТВ15 (74ALS109А) (JK -тригер з позитивним фронтом), побудувати тригер, еквівалентний КР1533ТМ2 (74ALS74А) (D -тригер), без застосування інших компонентів.

5. Використовуючи елемент КР1533ТМ2 (74ALS74А) (D -тригер з позитивним фронтом), побудувати тригер, еквівалентний КР1533ТВ15 (74ALS109А), без застосування інших компонентів.

6. На рис. 18 приведена схема RS -тригера на логічних елементах **2І-НІ**. Виконати наступні завдання: а) використовуючи допоміжні логічні елементи, розробити схему синхронного RS -тригера; б) використовуючи допоміжні ЛЕ, розробити схему синхронного RS -тригера з прямими входами S - і R -; в) використовуючи допоміжні ЛЕ, розробити схему розширювача тривалості імпульсу високого рівня; г) використовуючи допоміжні ЛЕ, розробити схему розширювача тривалості імпульсу низького рівня.

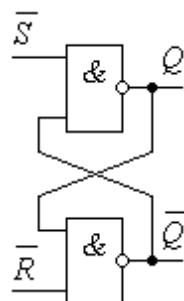


Рис. 18

7. На рис. 19 приводяться часові діаграми сигналів, що подаються на входи S - і R - RS -тригера, виготовленого на ЛЕ **2АБО-НІ**. Побудувати часову діаграму сигналу на прямому виході тригера.

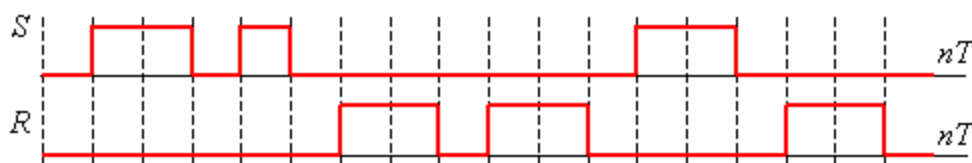


Рис. 19

8. Використовуючи RS -тригери і допоміжну логіку, розробити схему пристрою для зчитування і запам'ятовування (за керуючими сигналами Rd та Wr відповідно) $\frac{1}{2}$ байта інформації, що подається у паралельному форматі.

9. Послідовність синхроімпульсів подається одночасно на групу статичних синхронних тригерів, з'єднаних відповідно до рис. 20. Виконати наступні завдання: а) привести часові діаграми сигналів на кожному з виходів елементів Q_1 , Q_2 , Q_2 за умови, що тривалість синхросигналу набагато перевищує інтервал часу затримки на розповсюдження при нульових початкових значеннях Q_i ; б) побудувати часові діаграми сигналів на виходах Q_i за умови, що всі тригери динамічні і сприймають синхросигнал за його фронтом; в) побудувати часові діаграми сигналів на виходах Q_i за умови, що всі тригери динамічні і сприймають синхросигнал за його зрізом.

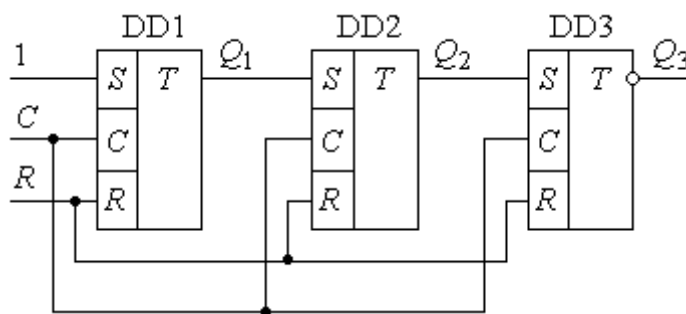


Рис. 20

10. На рис. 21 приводиться принципова схема цифрового пристрою з синхронними статичними RS -тригерами. Виконати наступне: а) пояснити роботу пристрою; б) побудувати часові діаграми напруг на виходах $Q_1 \dots Q_4$ при послідовній подачі чотирьох синхросигналів; в) записати логічні функції дешифратора для перетворення вихідного коду, який знімається з $Q_1 \dots Q_4$, у двійковий.

11. На рис. 22 приводиться схема двійкового лічильника від 0 до 4. При подачі чергового імпульсу сигнал високого рівня з'являється на відповідному виході. П'ятий імпульс приводить до обнуління схеми. Проаналізувати роботу лічильника. Побудувати часові діаграми сигналів на виходах логічних елементів $DD1 \dots DD5$; на виходах тригерів $Q_1 \dots Q_4$.

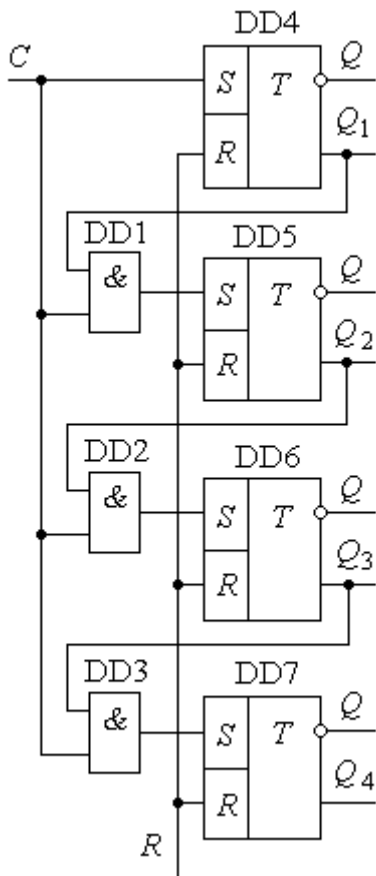


Рис. 21

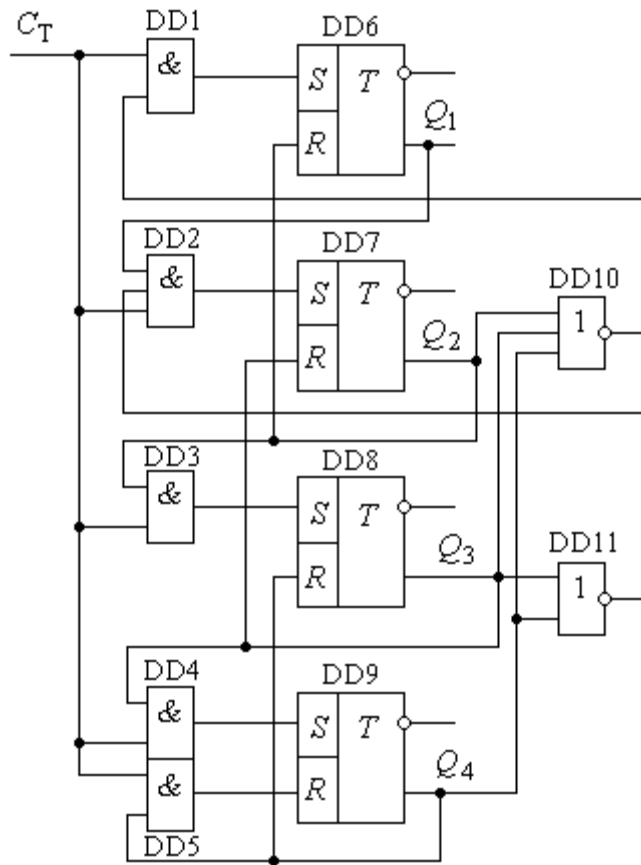


Рис. 22

12. Робота RS -тригера описується формулою $Q_{n+1} = S_n + Q_n \bar{R}_n$, $\bar{Q}_{n+1} = R_n + \bar{Q}_n \bar{S}_n$. Довести, що, використовуючи допоміжний інвертор DD1 (рис. 23), матимемо D -тригер, робота якого описується формулою $Q_{n+1} = C_n D_n + \bar{C}_n Q_n$.

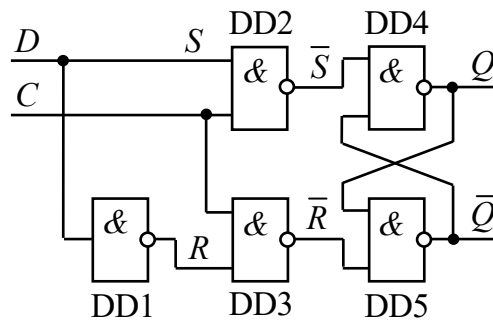


Рис. 23

13. На рис. 24 приведена схема двотактового динамічного тригера. Використовуючи часові діаграми, пояснити роботу тригера з урахуванням часових затримок на перемикання.

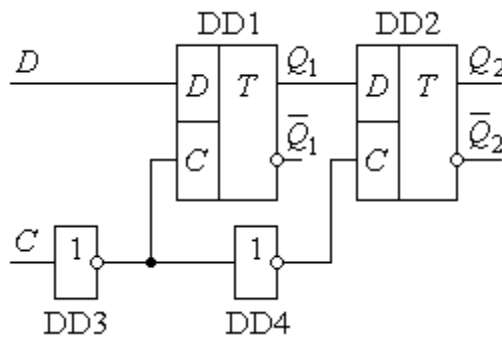


Рис. 24

14. На рис. 25 приведена схема D -тригера, в якому запис інформації забезпечується синхросигналом C_2 , а його обнуління – сигналом $\overline{C_1}$. Привести часові діаграми і пояснити його роботу.

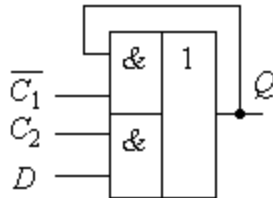


Рис. 25

15. Використовуючи алгоритм функціонування RS -тригера, розробити алгоритм функціонування пристрою, схема якого приведена на рис. 26, відносно входів x_1 та x_2 . Привести часові діаграми роботи пристрою.

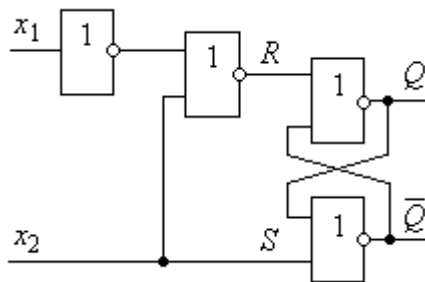


Рис. 26

16. Аналогічно до вправи 15, розробити алгоритм функціонування пристрою, схема якого приведена на рис. 27. Привести часові діаграми роботи пристрою.

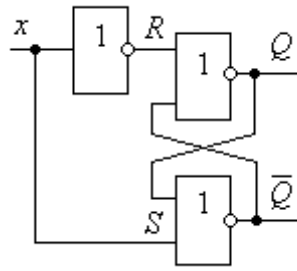


Рис. 27

17. Використовуючи синхронний RS -тригер і допоміжну логіку, розробити схему синхронного D -тригера.

18. Робота D -тригера Ерла описується формулою $Q_{n+1} = \overline{C_n Q_n} \cdot \overline{C_n D_n} \cdot \overline{D_n Q_n}$. Побудувати схему тригера у базисі **I-НІ**, пояснити її роботу.

19. Розробити тактований фронтом двовходовий тригер, що функціонує відповідно до табл. 9, на основі синхронного D -тригера.

Табл. 9

| x_1 | x_0 | Q_{n+1} |
|-------|-------|------------------|
| 0 | 0 | 0 |
| 0 | 1 | Q_n |
| 1 | 0 | $\overline{Q_n}$ |
| 1 | 1 | 1 |

20. Розробити синхронний двовходовий тригер, що функціонує відповідно до табл. 10, на базі універсального JK -тригера.

Таблиця 10

| x_1 | x_0 | Q_{n+1} |
|-------|-------|------------------|
| 0 | 0 | 0 |
| 0 | 1 | Q_n |
| 1 | 0 | $\overline{Q_n}$ |
| 1 | 1 | 0 |

21. Побудувати граф-схему JK -тригера. Пояснити, чим вона відрізнятиметься від граф-схеми RS -тригера.

22. На рис. 28 приведена схема тригера. Визначити тип тригера, пояснити принцип його роботи.

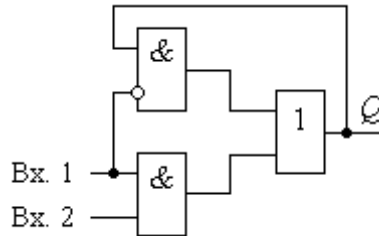


Рис. 28

23. На рис. 29 приведена схема тригера. Побудувати таблицю станів і описати алгоритм його функціонування по відношенню до сигналів x_0, x_1 .

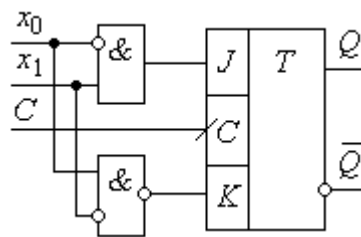


Рис. 29

24. На рис. 30 приведена схема старт-стопного пристрою. Виходячи з умови, що тривалість імпульсів “Старт” і “Стоп” достатня для зміни стану тригера, а період T набагато менший інтервалу часу між подачею сигналів “Старт” і “Стоп”, побудувати часові діаграми роботи пристрою.

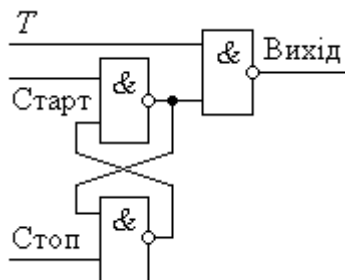


Рис. 30

25. Проаналізувати особливості роботи старт-стопних пристроїв, схеми яких приведені на рис. 31, *а – б*.

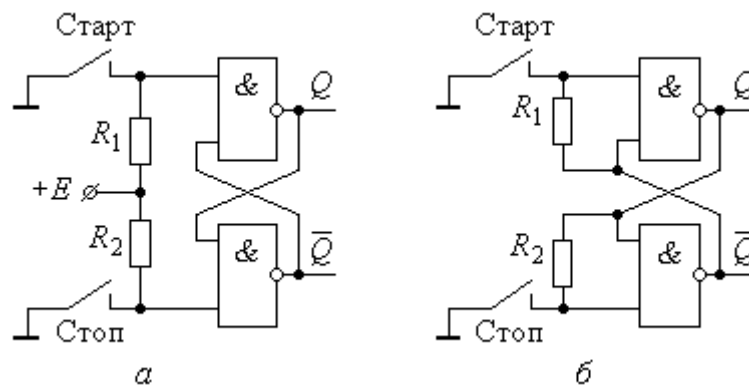


Рис. 31

26. На рис. 32 приводяться схеми старт-стопних пристроїв, взяті з літературних джерел (Преснухин Л.Н., Воробьєв Н.В., Шишкевич А.А. Расчёт элементов цифровых устройств. – М.: Высшая школа, 1991. – 326 с.). Проаналізувати особливості роботи пристроїв при різних часових співвідношеннях між сигналами “Старт” і “Стоп”.

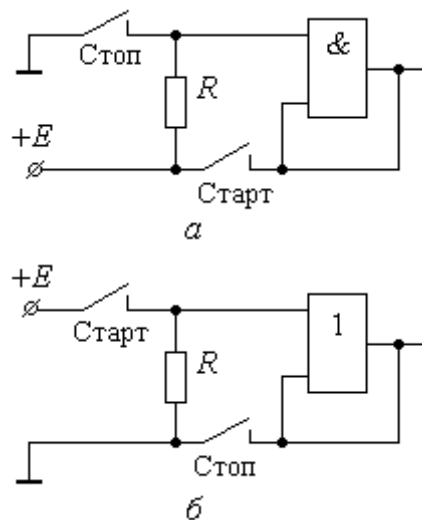


Рис. 32

27. Розробити варіанти схем старт-стопних пристроїв з використанням синхронних *D*-тригерів; *JK*-тригерів.

28. На входи *D* і *C* *D*-тригера подаються прямокутні імпульси з близькими частотами (наприклад, 90 Гц і 100 Гц). Пояснити особливості роботи тригера. Навести приблизний вигляд часових діаграм імпульсів на виході. Оцінити кількість імпульсів на виході тригера за інтервал 1 секунда.

29. На рис. 33, *а* приведена схема на основі двох *D*-тригерів, на входи якої поступають сигнали *T* і *C* відповідно до часової діаграми, приведеної на рис. 33, *б*. Пояснити роботу пристрою і його призначення. Побудувати часові діаграми сигналів на виходах Q_1 і Q_2 тригерів і на виході пристрою *Y* без урахування часових затримок на перемикання тригерів. Який вплив матимуть часові затримки на роботу схеми? Як зміняться часові діаграми, якщо тригер DD2 спрацюватиме не за зрізом, а за фронтом сигналу тактового генератора *T*?

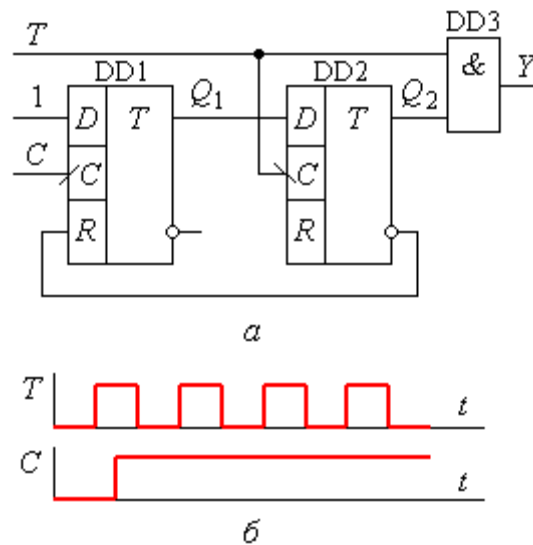


Рис. 33

30. Пояснити, як побудувати *T*-тригер, використовуючи *D*-тригер і, якщо потрібно, допоміжну логіку.

31. Пояснити, як побудувати *JK*-тригер, використовуючи *T*-тригер і, якщо потрібно, допоміжну логіку.

32. Використовуючи *D*-тригер і допоміжну комбінаційну логіку, розробити схему *T*-тригера з допоміжним дозволяючим входом.

33. Використовуючи *T*-тригер і допоміжну комбінаційну логіку, розробити схему *D*-тригера з допоміжним дозволяючим входом.

34. На рис. 34 приводяться часові діаграми *S*- і *R*- сигналів, що діють на *RS*-тригер, виготовлений з використанням логічних елементів **АБО-НІ**. Побудувати часові діаграми сигналу на прямому виході Q за умови, що у момент часу $t = 0$ стан виходу $Q_n = 0$.

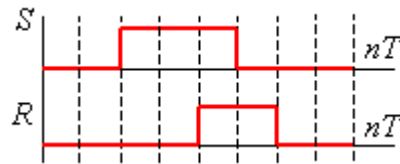


Рис. 34

35. На рис. 35 приводяться часові діаграми сигналів, що прикладаються до входів \bar{S} і \bar{R} RS-тригера, виготовленого з використанням ЛЕ І-НІ. Побудувати часові діаграми сигналу на прямому виході Q за умови, що у момент часу $t = 0$ стан виходу $Q_n = 0$.

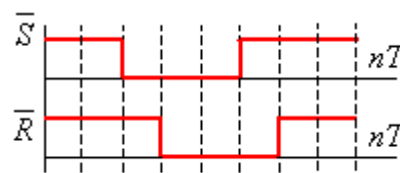


Рис. 35

Теоретичні відомості

Тригери в інтегральному виконанні будуються з простих логічних елементів типу АБО-НЕ, І-НЕ. У загальному випадку тригер складається з логічного пристрою керування та власне тригера як елемента пам'яті. Існує велика кількість різноманітних схем тригерів із різними функціональними можливостями. Узагальнену структурну схему тригера зображено на рис. 36.



Рис. 36

Тригер має два виходи: Q – прямий (одиничний), \bar{Q} – інверсний (нульовий). Входи A_i називаються інформаційними, а входи C_i – тактовими або синхронізуючими.

Стани тригера в таблицях переходів зазвичай вказують так:

- 0 – тригер має сигнал на виході $Q = 0$ (нульовий стан) незалежно від сигналів на входах;
- 1 – тригер має сигнал на виході $Q = 1$ (одиничний стан) незалежно від сигналів на входах;
- Q^i – стан тригера не змінюється при зміні сигналів на входах;
- \bar{Q} – стан тригера змінюється на протилежний при зміні сигналів на входах;
- X – невизначений стан тригера, коли він після зміни сигналів на входах рівноможливо може опинитися в нульовому ($Q = 0$) або у одиничному ($Q = 1$) стані.

Всі тригерні схеми розділяються на дві групи.

Перша з них – *flip-flop* – характеризується тим, що вибірка вхідних сигналів і відповідна зміна виходів визначається в моменти дії тактових часових сигналів (синхронні тригери).

Особливість другої групи схем – *latch* – полягає в тому, що вони змінюють свій стан при зміні вхідних сигналів незалежно від наявності чи відсутності часових тактових сигналів.

RS-тригери. Схема RS-тригера, зібраного на логічних елементах **2АБО-НІ**, приведена на рис. 37, *а*. На рис. 37, *б* приведено його умовне позначення.

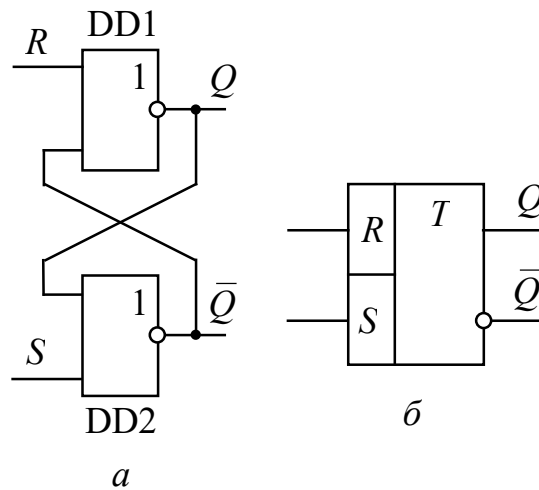


Рис. 37

Повна таблиця станів RS -тригера (Табл. 11) описує особливості його роботи.

Таблиця 11

| R_n | S_n | Q_n | Q_{n+1} |
|-------|-------|-------|-----------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | x |
| 1 | 1 | 1 | x |

При $S_n = R_n = 1$, незалежно від Q_n , стан виходів тригера є невизначеним. Розглянута комбінація вхідних сигналів називається *невизначеною комбінацією*, а перехід від такої комбінації до нульової – *забороненим переходом*.

Мінімізована таблиця станів тригера наведена у табл. 12.

Таблиця 12

| R_n | S_n | Q_{n+1} |
|-------|-------|-----------|
| 0 | 0 | Q_n |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | x |

D-тригери. Функціональна особливість тригерів цього типу полягає в тому, що сигнал на виході Q в $(n + 1)$ -ому такті повторює значення сигналу на вході D в n -му такті.

Таблиця 13

| C_n | D_n | Q_n | Q_{n+1} |
|-------|-------|-------|-----------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

| | | $C_n D_n$ | | | |
|-------|---|-----------|----|----|----|
| | | 00 | 01 | 11 | 10 |
| Q_n | 0 | 0 | 0 | 1 | 0 |
| | 1 | 1 | 1 | 1 | 0 |

Рис. 38

Роботу D -тригера пояснює таблиця його станів (табл. 13), з якої витікає, що при $C_n = 0$ значення $Q_{n+1} = Q_n$, а при $C_n = 1$ $Q_{n+1} = D_n$. Зобразивши логічну функцію $Q_{n+1} = f(Q_n, C_n, D_n)$ у вигляді карти Карно (рис. 38), після мінімізації знаходимо: $Q_{n+1} = C_n D_n + \overline{C_n} Q_n$. Враховуючи той факт, що друга складова цього рівняння характеризує лише режим зберігання інформації, закон функціонування D -тригера виразиться формулою: $Q_{n+1} = C_n D_n$.

Друге призначення D -тригера – зберігати дані (*Datas*), що надходять по D -входу. Синхросигнали в цьому випадку відіграють роль команди “Запис у тригер”. Особливість схеми проявляється в режимі $C = 1$. У цьому випадку будь-які зміни (t_5, t_6, t_7, t_8) на вході D приведуть до відповідних змін на виході Q . Тобто схема виконуватиме просто функцію повторювача сигналів, що присутні на вході D .

У деяких схемах вводять також V -вхід (від англ. *Valve* – клапан), що дає можливість вимикати тригер. Такі схеми називаються *DV-тригерами*.

Якщо в схемі синхронізованого імпульсом D -тригера замінити вхід D на інверсний \overline{D} , то виходи Q та \overline{Q} поміняються місцями. Така властивість може бути корисною при використанні мікросхем, в яких виведені лише прямі виходи (див. мікросхеми 555ІР22 та 1533ІР33).

JK-тригери. Цей тип тригерів за логікою роботи подібний до *RS*-тригерів, але, на відміну від них, не має невизначених переходів.

Скорочена таблиця станів (Табл. 14) пояснює наведену різницю, якщо прийняти, що *S*-вхід відповідає *J*-входу, а *R*-вхід – відповідно, *K*-.

Таблиця 14

| J_n | K_n | Q_{n+1} |
|-------|-------|------------------|
| 0 | 0 | Q_n |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | $\overline{Q_n}$ |

З таблиці витікає, що поява комбінації $J = K = 1$ у кожному такті призводить до зміни стану тригера на протилежний.

На відміну від *RS*-тригерів, схеми *JK*-тригерів виготовляються тільки синхронними.

Робота асинхронного *JK*-тригера описується наступним характеристичним рівнянням: $Q_{n+1} = J_n \overline{Q_n} + \overline{K_n} Q_n$. Для синхронного *JK*-тригера є справедливим рівняння: $Q_{n+1} = C(J_n \overline{Q_n} + \overline{K_n} Q_n) + \overline{C} Q_n$.

Контрольні запитання

1. До якого класу пристроїв ставляться тригери: до комбінаційного схем або цифровим автоматам й у чому основна відмінність?
2. Назвіть спосіб завдання законів функціонування тригерів.
3. По яких ознаках класифікуються тригерні пристрої?
4. Які відмінності синхронних від асинхронних тригерів?
5. У чому полягають переваги тригерів із затримкою (типу Master-slave)?
6. Функції яких тригерів може виконувати універсальний J-K тригер?
7. Призначення тригерів D-V типу.

Рекомендована література

1. Тиртишніков О.І., Корж Ю.М. Обчислювальна техніка та мікропроцесори. Частина 2. Цифрові автомати: Навчальний посібник. – Полтава: ПВІЗ, 2006, с. 34 – 46.
3. Калабеков Б.А., Мамзелев И.А. Цифровые устройства и микропроцессорные системы. М.: Радио и связь, 1987.

Практичне заняття № 11 – 15

Тема: Скінченні автомати.

Мета: навчитися проектувати цифрові автомати з пам'яттю.

Завдання

1. Проаналізувати роботу синхронного автомата, схема якого приведена на рис. 39. Скласти таблицю переходів, записати рівняння збудження, зобразити часову діаграму роботи автомата.

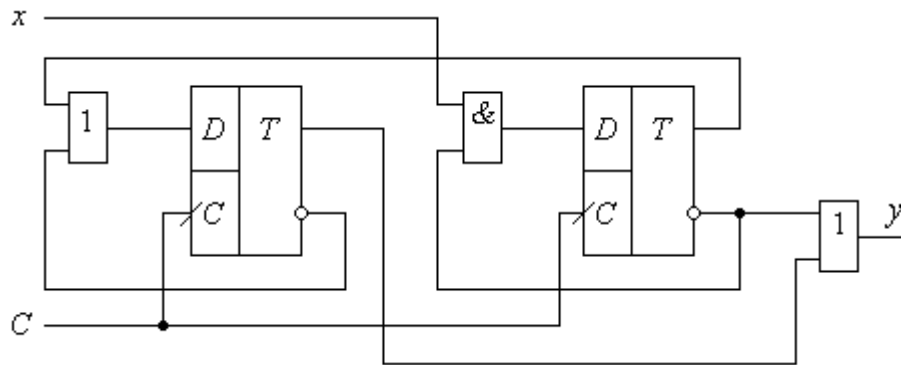


Рис. 39

2. Повторити завдання 1, замінивши місцями елементи **2АБО** та **2І**. Порівняти таблиці переходів, дати пояснення.

3. Побудувати граф переходів для скінченного автомата, що описується таблицею переходів (табл. 15).

Таблиця 15

| $Q \backslash X$ | $x_1 x_0$ | | | | Y |
|------------------|-----------|-------|-------|-------|-----|
| | 00 | 01 | 11 | 10 | |
| Q_0 | Q_1 | Q_1 | Q_2 | Q_2 | 0 |
| Q_1 | Q_3 | Q_3 | Q_2 | Q_2 | 0 |
| Q_2 | Q_1 | Q_1 | Q_4 | Q_4 | 0 |
| Q_3 | Q_3 | Q_3 | Q_4 | Q_2 | 1 |
| Q_4 | Q_1 | Q_3 | Q_4 | Q_4 | 1 |

4. Побудувати граф-схему автомата, заданого таблицею переходів (табл. 16).

Таблиця 16

| $Q \backslash X$ | $x_1 x_0$ | | | | Y |
|------------------|-----------|-------|-------|-------|-----|
| | 00 | 01 | 11 | 10 | |
| Q_0 | Q_0 | Q_1 | Q_2 | Q_1 | 1 |
| Q_1 | Q_1 | Q_2 | Q_3 | Q_2 | 0 |
| Q_2 | Q_2 | Q_3 | Q_0 | Q_3 | 0 |
| Q_3 | Q_3 | Q_0 | Q_1 | Q_0 | 0 |

5. Побудувати граф-схему автомата, заданого таблицею переходів (табл. 17).

Таблиця 17

| $Q \backslash X$ | x | |
|------------------|-----|-----|
| | 0 | 1 |
| A | B | A |
| B | B | C |
| C | B | D |
| D | E | A |
| E | B | F |
| F | B | G |
| G | E | H |
| H | B | A |

6. Розробити таблицю переходів і виходів скінченного автомата, заданого граф-схемою, що приведена на рис. 40. Під позначеннями станів $Q_0 \div Q_7$ вказані значення виходів $y_1 y_2$ автомата.

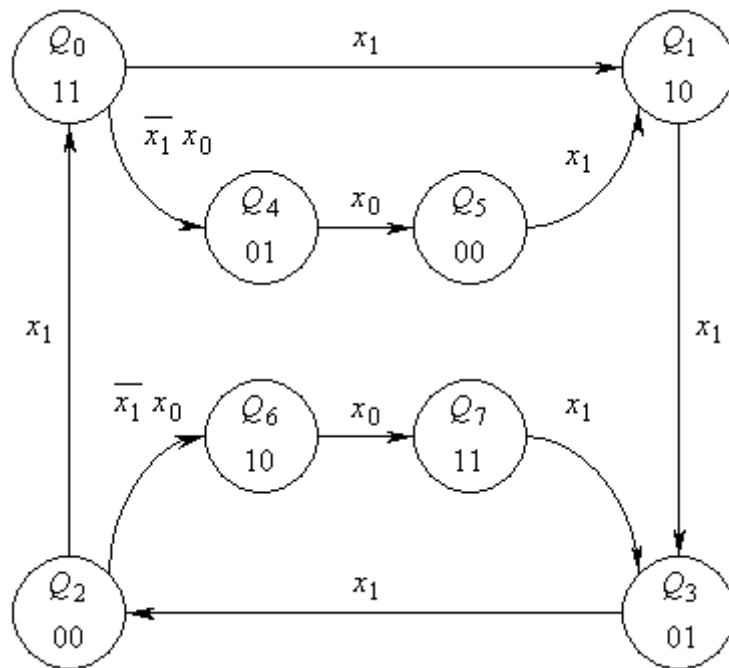


Рис. 40

7. Виконати аналіз роботи синхронного автомата, функціональна схема якого приводиться на рис. 41. Побудувати часові діаграми його роботи, записати рівняння збудження D -тригерів, скласти таблицю переходів і граф-схему переходів автомата.

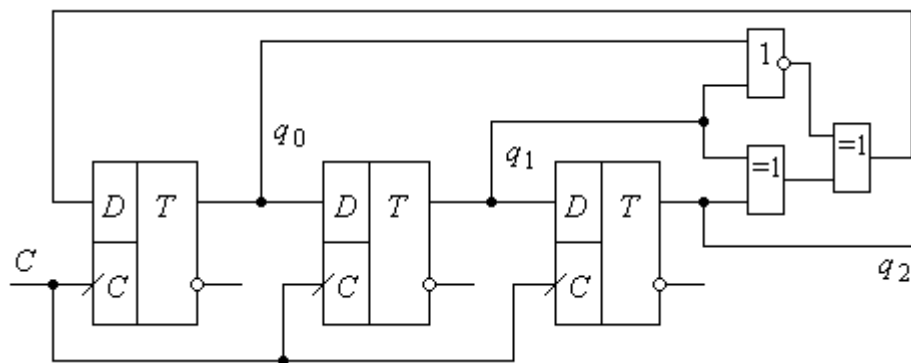


Рис. 41

8. За аналогією з вправою 7, виконати аналіз роботи синхронного автомата, функціональна схема якого приведена на рис. 42.

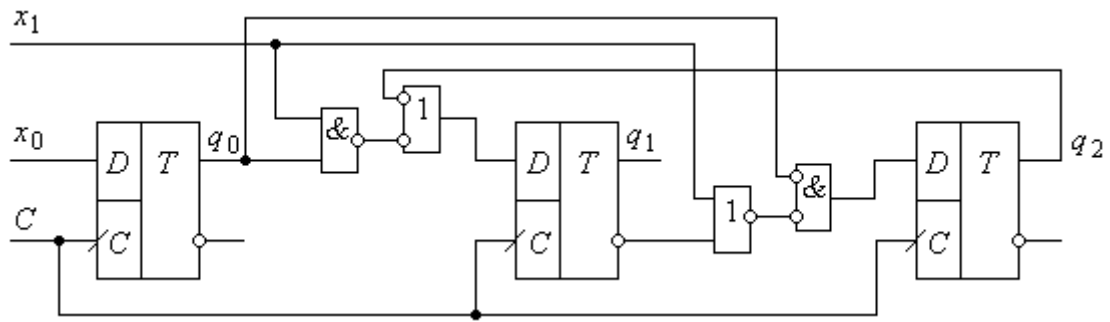


Рис. 42

9. За аналогією з вправою 7, виконати аналіз роботи синхронного автомата, функціональна схема якого приведена на рис. 43. Аналіз провести для $x=1$, починаючи з нульового початкового стану $Q_0 = q_1 q_0 = 00$.

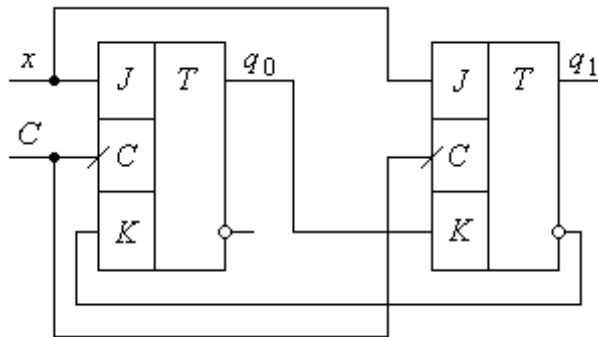


Рис. 43

10. За аналогією з вправою 7, виконати аналіз роботи синхронного скінченного автомата, функціональна схема якого приведена на рис. 44. Аналіз виконати при умові $x=1$.

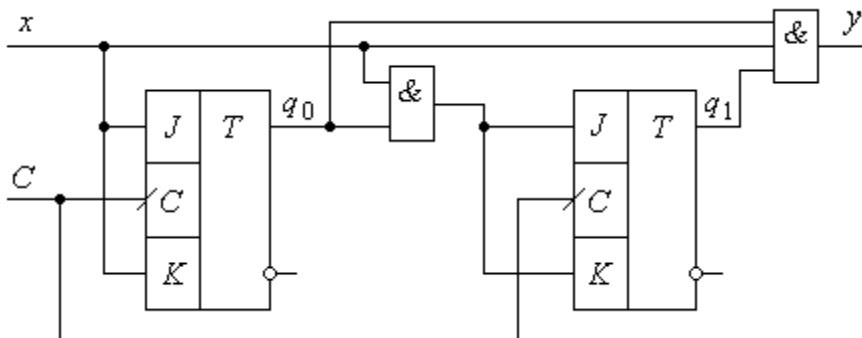


Рис. 44

11. За аналогією з попередніми вправами, виконати аналіз синхронного скінченного автомата, функціональна схема якого приведена на рис. 45. Автомат виготовлений на T -тригерах з дозволяючими входами L -.

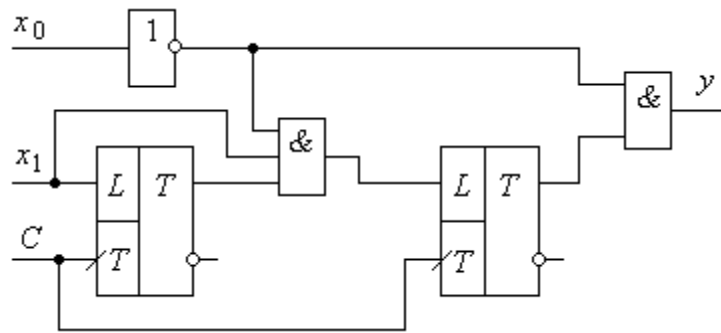


Рис. 45

12. Спроекувати синхронний автомат Мура і описати його роботу за допомогою часових діаграм. Автомат має два входи x_1 та x_0 і один вихід y . Як тільки вхід x_0 встановиться в **1**, на виході y з'явиться низький рівень вихідного сигналу ($y=0$). Якщо x_0 змінить своє значення в **0**, то y залишатиметься на низькому рівні при $x_1=0$ протягом двох тактів і перейде в **1** на наступних двох тактах. При $x_1=1$ автомат відразу перейде в стан $y=1$ і залишатиметься у такому стані до моменту часу, поки x_0 знову не встановиться в **1**.

13. Спроекувати синхронний тактований автомат Мура, який виконував би перевірку лінії послідовних даних, що приймаються, на парність. Автомат повинен мати два входи – вхід синхронізації C - і вхід даних D -, а також вхід тактового генератора, і один вихід y , високий рівень якого вказує на помилку в передачі даних. В автоматі потрібно забезпечити можливість перевірки слова довжиною в 1 байт. В схемі можна використовувати D - або JK -тригери.

14. Спроекувати синхронний тактований автомат Мура, який виконував би перевірку лінії даних, що передаються, на парність. Автомат повинен мати два входи – вхід синхронізації C - і вхід даних D -, а також вхід тактового генератора, і один вихід y , високий рівень якого встановлює дев'ятий біт в **1**, якщо кількість одиниць в однобайтовому слові непарна. При непарній кількості одиниць $y=0$. В схемі можна використовувати D - або JK -тригери.

15. Спроектувати тактований синхронний автомат на основі таблиці переходів (табл. 18). Для синтезу автомата використовуйте D -тригери.

Таблиця 18

| $Q \backslash X$ | x | | y |
|------------------|-------|-------|-----|
| | 0 | 1 | |
| Q_0 | Q_1 | Q_3 | 0 |
| Q_1 | Q_2 | Q_1 | 0 |
| Q_2 | Q_1 | Q_0 | 1 |
| Q_3 | Q_1 | Q_2 | 0 |

16. Повторити завдання **15** з використанням JK -тригерів.

17. Виконати аналіз автомата, функціональна схема якого приведена на рис. 46. Скласти таблицю переходів, побудувати граф-схему та часові діаграми його роботи.

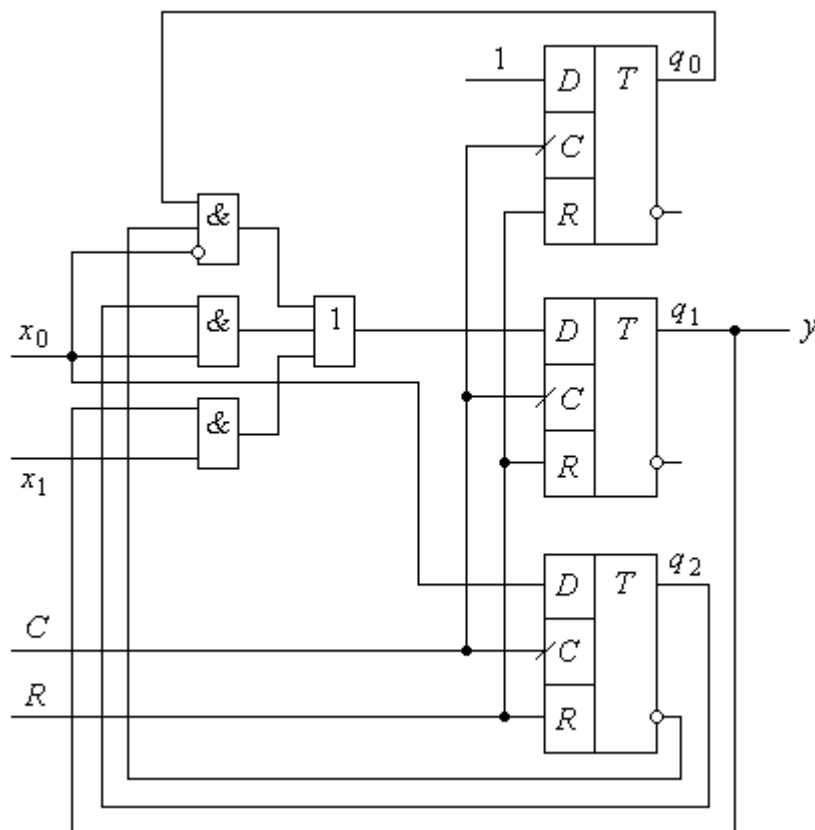


Рис. 46

18. Виконати синтез автомата за результатами завдання **17** з використанням JK -тригерів.

19. Таблиця 18 описує автомат, в якому початковий стан Q_0 призначений для його коректного запуску. Якщо цей стан видалити, автомат також працюватиме коректно, якщо керуючі сигнали будуть подані після встановлення перехідних процесів при подачі живлення. Синтезувати новий автомат з використанням D -тригерів зі скороченою кількістю станів.

20. Виконати завдання 19 з використанням JK -тригерів.

21. Розробити схему пристрою для виконання операції додавання двох чотирьохбітних слів у послідовному форматі з запам'ятовуванням сигналу переносу на один такт.

22. Розробити схему кодового замка (асинхронного автомата), який спрацює після правильного послідовного натискання чотирьох цифр.

23. З використанням RS -, D - і JK -тригерів розробити схеми цифрових автоматів для порівняння двох двійкових чисел у послідовному форматі. Порівнювані числа $X (x_{n-1}, \dots, x_1, x_0)$ та $Y (y_{n-1}, \dots, y_1, y_0)$. При порівнянні необхідно встановити їх співвідношення: $X > Y$; $X < Y$; $X = Y$, які описуються за допомогою функцій:

$$F_1 (X > Y) = \begin{cases} 1 & \text{при } X > Y; \\ 0 & \text{при } X \leq Y; \end{cases}$$

$$F_2 (X < Y) = \begin{cases} 1 & \text{при } X < Y; \\ 0 & \text{при } X \geq Y; \end{cases}$$

$$F_3 (X = Y) = \begin{cases} 1 & \text{при } X = Y; \\ 0 & \text{при } X \neq Y. \end{cases}$$

24. Розробити асинхронний автомат для сортування деталей **A**, **B**, **C**, що переміщуються на конвеєрі у довільному порядку.

25. Задача про подачу води в водонапірну вежу. Два насоси ρ_1 і ρ_2 використовуються для подачі води в водонапірну вежу. Обидва починають працювати, коли вода опуститься нижче рівня мітки 1, і працюють до того часу,

поки рівень води не підніметься до мітки 2. При цьому насос ρ_1 зупиниться, а насос ρ_2 продовжуватиме роботу до того часу, поки рівень води не підніметься до мітки 3, після чого він зупиниться. Запуститися обидва насоси повинні тоді, коли вода опуститься до мітки 1. Розробити автомат керування насосами.

26. Задача визначення кількості відвідувачів музею. Спроектувати автомат, який повинен підрахувати кількість відвідувачів музею за визначений інтервал часу. В якості датчика використовуються дві оптопари O_1 та O_2 , рознесені на відстань 1 м. Відвідувач фіксується, якщо він перетне спочатку промінь першої оптопари, а потім другої. Зворотний процес не враховується.

Теоретичні відомості

Цифрові пристрої з m тригерами ($m > 1$), стан виходів яких залежить не тільки від значень вхідних сигналів у заданий момент часу, а й від стану використовуваних тригерів у даний та попередній моменти часу, називаються *цифровими автоматами*.

Скінченні автомати можуть бути *синхронними*, зміна станів яких відбувається в тактові моменти часу, що задаються зовнішнім генератором. В *асинхронних* автоматах зміна станів відбувається внаслідок дії вхідних сигналів без затримок. Тому асинхронні автомати вважаються більш швидкодіючими і знаходять використання у швидкодіючих інформаційних пристроях вимірювання і керування різноманітними процесами, де необхідна миттєва реакція на зміну вхідних сигналів.

Синхронні автомати в силу специфіки своєї роботи вносять у процес вимірювання або керування затримку, що визначається величиною періоду синхросигналу.

Здебільшого асинхронні автомати будуються на основі асинхронних елементів пам'яті – асинхронних тригерів. Синхронні автомати будуються з використанням синхронних тригерів.

Автомати Мілі і Мура. На практиці використання цифрових автоматів можна виділити невелику кількість типових алгоритмів їх функціонування. Найбільшого розповсюдження набули два типи автоматів – *автомати Мілі і Мура*.

Закон функціонування автомата Мілі задається рівняннями:

$$\begin{cases} Q(t+1) = f_p [Q(t), X(t)]; \\ Y_k(t) = \lambda [Q(t), X(t)]; \quad t = 0, 1, 2, \dots \end{cases} \quad (11.1)$$

Закон функціонування автомата Мура описується рівняннями:

$$\begin{cases} Q(t+1) = f_p [Q(t), X(t)]; \\ Y_k(t) = \lambda [Q(t)]; \quad t = 0, 1, 2, \dots \end{cases} \quad (11.2)$$

Різниця між автоматами Мілі і Мура полягає лише в тому, що вихідний сигнал останнього залежить тільки від внутрішнього стану, у той час як в автоматі Мілі значення вихідного сигналу залежить також і від вхідного сигналу.

Структура автомата Мілі зображена на рис. 47.

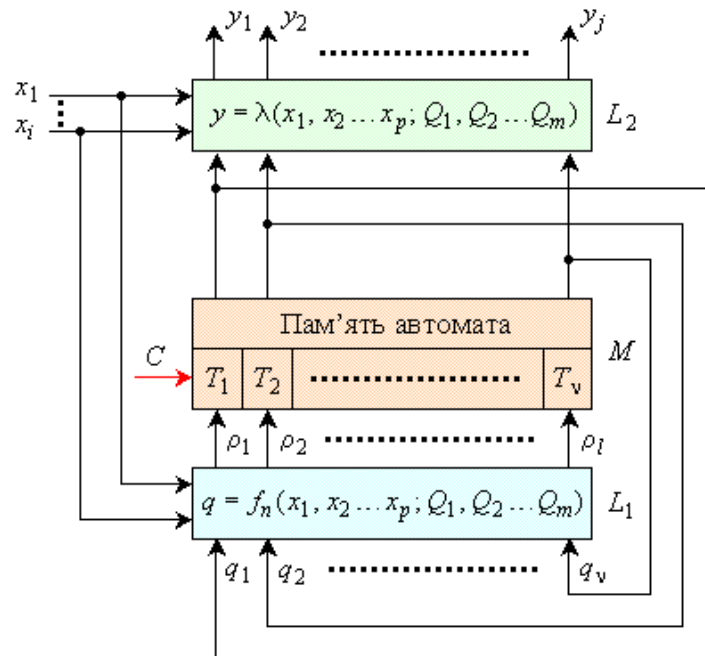


Рис. 47

Автомат складається з набору елементарних автоматів (тригерів T_1, T_2, \dots, T_n), стани яких q_1, q_2, \dots, q_n в кожний момент часу визначають внутрішній

стан автомата в цілому. Під дією вхідних сигналів $x_1 \dots x_i$, які подаються в дискретні моменти часу, відбувається формування сигналів $\rho_1 \dots \rho_l$, що забезпечують послідовне перемикання тригерів $T_1 \dots T_n$. Таким шляхом відбувається послідовна зміна станів автомата. Оскільки комбінаційна схема L_1 формує керуючі сигнали тригерів в залежності від значень виходів тригерів у момент подачі тактового сигналу (для синхронних схем) або в моменти подачі вхідних сигналів (для асинхронних), то значення виходів тригерів у послідуєчий момент часу i , відповідно, стан автомата однозначно визначається вхідними сигналами і станом автомата в попередній момент часу.

Сигнали керування тригерами $\rho_1 \dots \rho_l$ формуються комбінаційним пристроєм L_1 , структура якого визначає функцію переходів автомата. Функція виходів реалізується комбінаційним пристроєм L_2 , який формує сигнали $y_1 \dots y_j$ як функції виходів пам'яті (для автоматів Мура) або як функції виходів пам'яті і вхідних сигналів (для автоматів Мілі).

Переходячи до відносного часу роботи, формули (11.1), (11.2) можна зобразити у вигляді:

- для автомата Мілі:

$$\begin{cases} Y_{n+1} = L_2(Q_{n+1}, X_n); \\ Q_{n+1} = L_1(Q_n, X_n); \end{cases} \quad (11.3)$$

- для автомата Мура:

$$\begin{cases} Y_{n+1} = L_2(Q_{n+1}); \\ Q_{n+1} = L_1(Q_n, X_n). \end{cases} \quad (11.4)$$

Рівняння (11.3), (11.4) можуть бути заданими аналітично або у вигляді таблиць станів (таблиць відповідностей).

Взаємозв'язок між поточними Q_n і послідуєчими значеннями Q_{n+1} виходів визначається характеристичними рівняннями використовуваних тригерів.

Поєднуючи характеристичні рівняння тригерів і рівняння комбінаційних схем, можна проводити аналіз роботи існуючого автомата або виконувати його синтез.

Способи опису роботи автоматів. В практиці аналізу і синтезу цифрових автоматів використовують різні способи опису їх роботи. Найбільш поширеними є *табличний і графічний способи*.

Розглянемо спочатку опис роботи автомата з використанням таблиць переходів та виходів. Стовбці (рядки) цих таблиць позначають символами з множини Q , а рядки (стовбці) – символами з множини X .

Кількість рядків таблиці переходів визначається кількістю комбінацій вхідних сигналів ρ , а кількість стовбців – відповідно, кількість станів M автомата.

У табл. 20 зображена таблиця автомата з $\rho = 3$, $M = 2^m = 4$.

Таблиця 20

| $X \backslash Q$ | Q_0 | Q_1 | Q_2 | Q_3 |
|------------------|-------|-------|-------|-------|
| X_1 | Q_3 | Q_2 | Q_1 | Q_0 |
| X_2 | Q_0 | Q_3 | Q_2 | Q_1 |
| X_3 | Q_1 | Q_0 | Q_3 | Q_2 |

В кожній клітці таблиці переходів записується стан, в який переходить автомат з попереднього стану (стану, що стоїть у заголовку стовпця) при дії відповідного вхідного сигналу. Так, наприклад, якщо автомат знаходиться у стані Q_0 , то при дії сигналу X_1 він перейде в стан Q_3 ; при дії сигналу X_2 залишиться в стані Q_0 , а при дії сигналу X_3 перейде в стан Q_1 .

Таблиця виходів (табл. 21) відрізняється від таблиці переходів лише тим, що у кожній клітці записується відповідне значення вихідного сигналу автомата.

Таблиця 21

| $X \backslash Q$ | Q_0 | Q_1 | Q_2 | Q_3 |
|------------------|-------|-------|-------|-------|
| X_1 | Y_0 | Y_1 | Y_2 | Y_3 |
| X_2 | Y_3 | Y_0 | Y_1 | Y_2 |
| X_3 | Y_2 | Y_3 | Y_0 | Y_0 |

Таблиці переходів і виходів автомата Мілі можуть бути представлені у вигляді однієї суміщеної таблиці, у клітках якої вказані значення як станів, так і виходів.

Функції переходів і виходів автомата Мура задаються однією таблицею переходів, яка будується так само, як і таблиця переходів автомата Мілі. Різниця полягає лише в тому, що над заголовками кожного стовпця встановлюється окремим рядком значення виходів автомата (табл. 22).

Таблиця 22

| | | Y_2 | Y_1 | Y_0 |
|------------------|-------|-------|-------|-------|
| $X \backslash Q$ | Q | Q_0 | Q_1 | Q_2 |
| X_1 | Q_2 | Q_1 | Q_2 | |
| X_2 | Q_0 | Q_2 | Q_1 | |
| X_3 | Q_1 | Q_0 | Q_0 | |

Для частково заданих автоматів, у яких функції виходів і функції переходів визначені не для всіх комбінацій Q_m і X_p , відповідні клітки залишаються незаповненими.

В тому випадку, коли X_1 є тактовим сигналом, що діє лише на тригери, а стани автомата змінюються упорядковано в зростаючому або спадаючому напрямку, автомат називається *лічильним автоматом*, або *лічильником*.

Більш наочним є спосіб опису автоматів з допомогою графів, подібно до того, як описувалася робота тригерів. Різниця полягає в тому, що автомат може мати суттєво більшу кількість станів. На рис. 48 показані граф-схеми автоматів Мілі і Мура, які задані табл. 20 – 21 (рис. 48, а), Табл.22 (рис. 48, б).

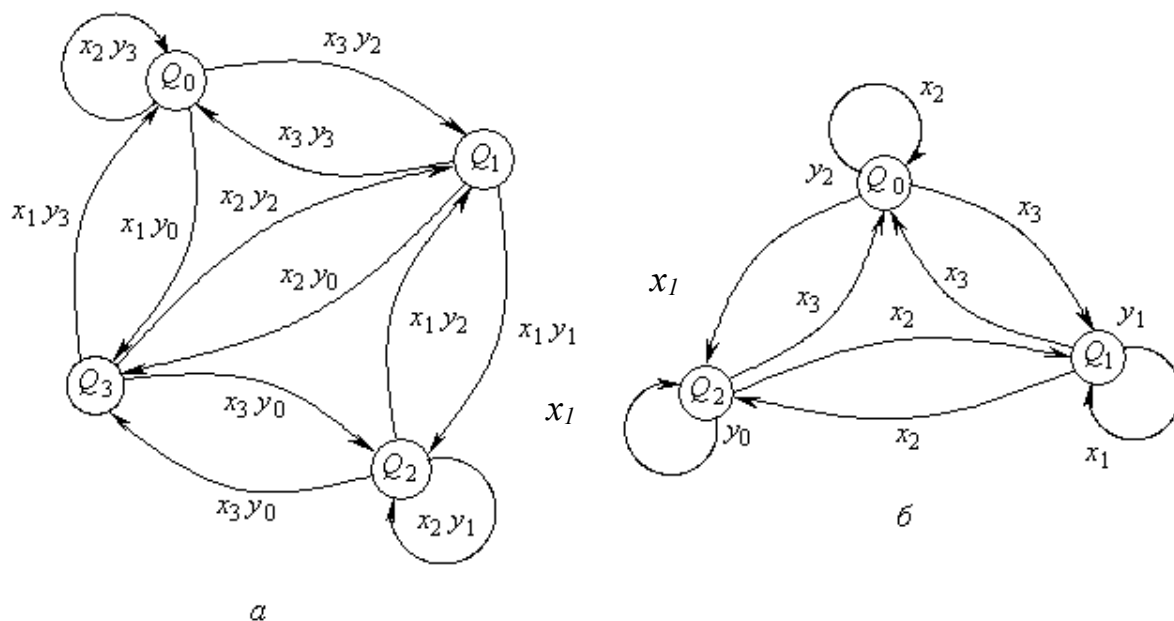


Рис. 48

Граф-схеми широко використовуються як при аналізі, так і при синтезі автоматів, а також при переході від словесного до формалізованого їх опису.

За допомогою таблиць переходів і виходів, як і за допомогою граф-схеми завжди можна знайти вихідну реакцію автомата на будь-яке вхідне слово, що належить множині X .

Однією з форм зображення автомата є його *дерево переходів і виходів*. Дерево може мати декілька ярусів, у кожному з яких за допомогою гілок показуються можливі переходи, починаючи з нульового стану.

На рис. 49 приводиться приклад дерева переходів, що відповідає табл. 20.

Еквівалентні перетворення автоматів. Для будь-якого автомата Мілі можна побудувати відповідний автомат Мура, і навпаки. Розглянемо алгоритми таких переходів на прикладі автомата Мура, що заданий табл. 23 і табл. 24.

Таблиця 23

| $X \backslash Q$ | Q_0 | Q_1 | Q_2 | Q_3 |
|------------------|-------|-------|-------|-------|
| x_1 | Q_1 | Q_2 | Q_0 | Q_2 |
| x_2 | Q_3 | Q_0 | Q_0 | Q_0 |
| x_3 | Q_2 | Q_3 | Q_2 | Q_3 |

Таблиця 24

| $X \backslash Q$ | Q_0 | Q_1 | Q_2 | Q_3 |
|------------------|-------|-------|-------|-------|
| x_1 | Y_3 | Y_2 | Y_1 | Y_1 |
| x_2 | Y_2 | Y_1 | Y_3 | Y_3 |
| x_3 | Y_1 | Y_2 | Y_2 | Y_3 |

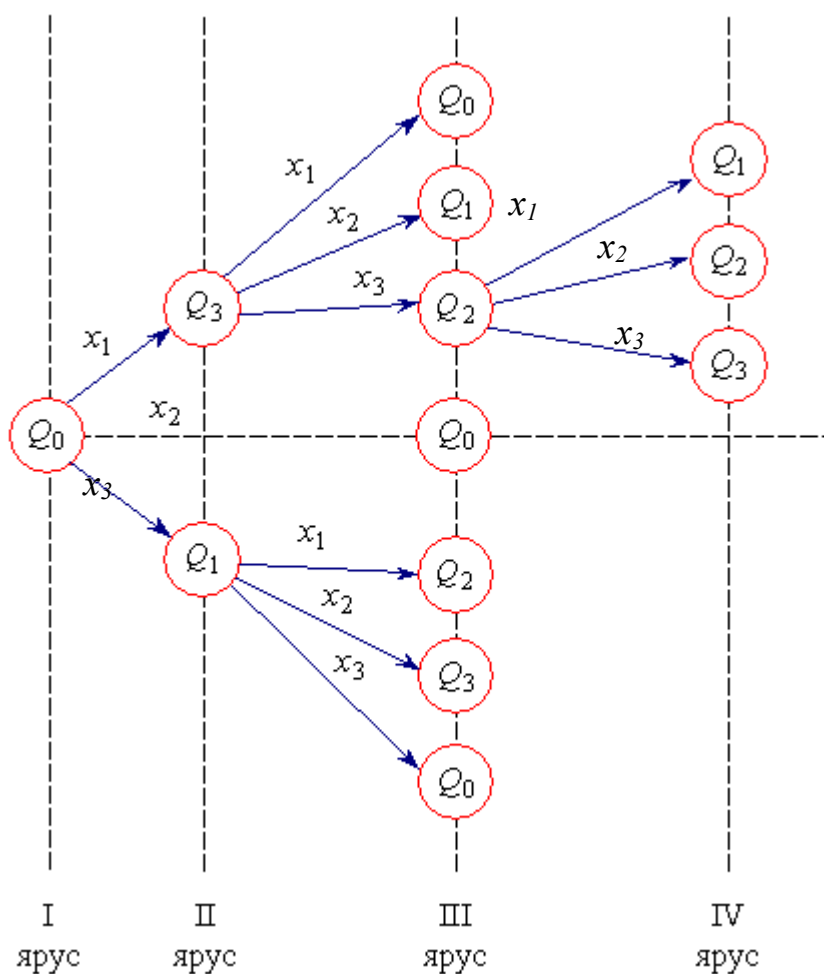


Рис. 49

Поставимо у відповідність кожній парі Q_m та X_p автомата Мілі стан Q_{mp} автомата Мура. До множини станів автомата Мура включимо початковий стан автомата Мілі Q_0 , але позначимо його Q_{00} . Для прикладу, що розглядається, така відповідність приведена в табл. 25.

Таблиця 25

| $X \backslash Q$ | Q_0 Q_{00} | Q_1 | Q_2 | Q_3 |
|------------------|-------------------|-------------------|-------------------|-------------------|
| x_1 | Q_1 Q_{01} | Q_2 Q_{11} | Q_0 Q_{21} | Q_2 Q_{31} |
| x_2 | Q_3 Q_{02} | Q_0 Q_{12} | Q_0 Q_{22} | Q_0 Q_{32} |
| x_3 | Q_2 Q_{03} | Q_3 Q_{13} | Q_2 Q_{23} | Q_3 Q_{33} |

Якщо автомат Мілі має m станів і p вхідних сигналів, то відповідний йому автомат Мура матиме $mp+1$ станів. З табл. 25 витікає той факт, що стан Q_0 автомата Мілі співпадає зі станами $Q_{00}, Q_{21}, Q_{12}, Q_{22}, Q_{32}$ автомата Мура. Тобто має місце наступна тотожність: $Q_0 = Q_{00} = Q_{21} = Q_{12} = Q_{22} = Q_{32}$. Аналогічно, $Q_1 = Q_{01}; Q_2 = Q_{03} = Q_{11} = Q_{23} = Q_{31}; Q_3 = Q_{02} = Q_{13} = Q_{33}$.

Така відповідність означає, що одному переходу автомата Мілі з Q_0 в Q_1 повинні бути відповідними всі переходи автомата Мура зі станів $Q_{00}, Q_{21}, Q_{12}, Q_{22}, Q_{32}$ в стан Q_{01} ; переходу з Q_1 в Q_2 повинні бути відповідні всі переходи автомата Мура зі стану Q_{01} в стани $Q_{03}, Q_{11}, Q_{23}, Q_{31}$, і т.д. Зі сказаного витікає наступний висновок: якщо стан Q_{mp} входить до множини станів, що співпадає зі станом, наприклад, Q_1 , то стовбець таблиці переходів для стану Q_{mp} співпадатиме з стовбцем таблиці переходів для стану Q_1 . Значення m функції виходів для еквівалентного автомата Мура: $\lambda(Q_{mp}) = \lambda(Q_m, X_p)$ при $Q_{mp} \neq Q_{00}$.

Для початкового стану Q_{00} значення вихідного сигналу вибирається довільно.

Внаслідок описаних відповідностей і перетворень можна побудувати таблицю переходів і виходів еквівалентного автомата Мура (табл. 26).

Таблиця 26

| | Y_3 | Y_3 | Y_2 | Y_1 | Y_2 | Y_1 | Y_2 | Y_1 | Y_3 | Y_2 | Y_1 | Y_3 | Y_3 |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| | Q_{00} | Q_{01} | Q_{02} | Q_{03} | Q_{11} | Q_{12} | Q_{13} | Q_{21} | Q_{22} | Q_{23} | Q_{31} | Q_{32} | Q_{33} |
| X_1 | Q_{01} | Q_{11} | Q_{31} | Q_{21} | Q_{21} | Q_{01} | Q_{31} | Q_{01} | Q_{01} | Q_{21} | Q_{21} | Q_{01} | Q_{31} |
| X_2 | Q_{02} | Q_{12} | Q_{02} | Q_{22} | Q_{22} | Q_{02} | Q_{32} | Q_{02} | Q_{02} | Q_{22} | Q_{22} | Q_{12} | Q_{32} |
| X_3 | Q_{03} | Q_{13} | Q_{33} | Q_{23} | Q_{23} | Q_{03} | Q_{33} | Q_{33} | Q_{33} | Q_{23} | Q_{23} | Q_{33} | Q_{33} |

Якщо f_{p1} і λ_k функції переходів і виходів автомата Мура, то функції переходів $f_{p2}(Q, X)$ і виходів $\lambda_2(Q, X)$ еквівалентного автомата Мілі: $f_{p2}(Q, X) = f_{p1}(Q, X); \lambda_2(Q, X) = \lambda_1(f_{p1}(Q, X))$.

Звідси витікає, що таблиця переходів еквівалентного автомата Мілі співпадає з таблицею переходів автомата Мура, а в кожен клітку таблиці виходів записується символ, яким відмічений стан автомата Мура в заданій клітці.

При такому перетворенні граф автомата Мілі відрізняється від графу автомата Мура лише тим, що вихідні сигнали з вузлів графа перенесені на всі гілки, що входять у даний вузол.

Основи аналізу цифрових автоматів. Аналіз роботи автомата виконується з метою визначення його стану в послідовний тактовий момент часу і передбачення послідовних станів. Такі задачі з'являються при вивченні роботи невідомих схем, при налагодженні пристроїв цифрової схемотехніки. Задачі аналізу є досить складними і розв'язуються поетапно.

Рекомендуються наступні кроки:

- визначаються стани на послідовному тактовому моменті часу і значення виходів вхідної та вихідної комбінаційних функцій L_1 і L_2 ;
- використовуються функції L_1 і L_2 для побудови таблиці станів, яка повністю визначає послідовний стан і значення виходу схеми для кожної комбінації поточного стану і виходів;
- будується діаграма станів, яка містить інформацію з попереднього кроку (граф переходів).

Основи синтезу скінченних автоматів. Задача синтезу (проектування) скінченних автоматів майже діаметрально протилежна попередній задачі аналізу.

У ряді літературних джерел приводиться послідовність кроків, які поетапно необхідно робити в процесі розробки автомата. Приблизно такі кроки можна описати як етапи проектування:

1. Задається закон функціонування автомата;
2. Мінімізується кількість внутрішніх станів автомата;
3. Кодуються стани автомата;

4. Визначаються функції збудження елементів пам'яті і функції виходів, а також забезпечується їх мінімізація;

5. Складається функціональна схема автомата в вибраному елементному базисі.

Перші три етапи відносяться до *абстрактного синтезу* автомата, при якому по значеннях вхідних і вихідних сигналів визначаються і кодуються його стани.

При абстрактному синтезі на основі словесного, табличного, за допомогою часових діаграм і інших засобів опису роботи автомату виявляється закон його функціонування і задаються функції переходів і виходів. На цьому етапі забезпечується мінімізація кількості станів і визначається кількість елементів пам'яті, необхідних для побудови автомата.

Решта станів забезпечують процедуру одержання структурних схем автоматів, що задаються кодованими таблицями входів і виходів. Така процедура називається *структурним синтезом*.

Задача структурного синтезу полягає у виборі типів елементарних автоматів (тригерів) і знаходження такого способу їх з'єднання між собою, який забезпечить функціонування автомата у відповідності до заданих кодованими таблицями переходів і виходів.

Основною задачею цього етапу є синтез комбінаційних схем з урахуванням вибраних тригерів, знаходження мінімальних форм функцій збудження і виходів. Етап закінчується побудовою функціональної схеми автомата.

Контрольні питання

1. Дайте визначення цифрового автомату.
2. Нарисуйте структурну схему автомата Мілі.
3. Нарисуйте структурну схему автомата Мура.

4. Чим відрізняється цифровий автомат Мілі від цифрового автомата Мура.
5. З яких етапів складається структурний синтез цифрових автоматів.

Рекомендована література

1. Самофалов К.Г., Романкевич А.М., та ін Прикладна теорія цифрових автоматів. - Київ. «Вища школа» 1987.
2. Соловйов Г.М. Арифметичні пристрої ЕОМ. - М. «Енергія». 1978.
3. Савельєв А.Я. Прикладна теорія цифрових автоматів - М. «Вища школа». 1987.
4. Каган Б.М. Електронні обчислювальні машини і системи. - М. Вища школа. 1985.
5. Лисіков Б.Г. Арифметичні та логічні основи цифрових автоматів. - Мінськ. «Вишэйшая школа». 1980.

Практичне заняття № 16

Тема: Лічильники.

Мета: вивчення принципів побудови, логіки роботи і синтезу лічильників.

Завдання

1. Скільки тригерів необхідно для побудови двійкового послідовного асинхронного лічильника з $M = 1024$?
2. На вхід двійкового асинхронного додаючого лічильника з $M = 16$ подано $N = 87$ імпульсів. Який код встановиться на виходах Q лічильника?
3. Повторити завдання 2, але для віднімаючого лічильника з початковою установкою **0000**.
4. Розробити функціональну схему додаючого двійкового лічильника з $M = 4$, в якому стан його виходів визначається кількістю записаних в ньому одиниць.
5. Використовуючи мікросхему ТТЛ ІЕ5, розробити схему лічильника з коефіцієнтом перерахунку **10**.
6. Використовуючи мікросхему ІЕ5 і мультиплексор КП7 ТТЛ, розробити схему для перетворення однобайтового паралельного формату даних у послідовний з паузою в 2 біти між словами.
7. Спроекувати синхронні лічильники за модулем **12** на основі довільно взятої логіки та наступних елементів: а) T -тригерів; б) RS -тригерів; в) JK -тригерів; г) D -тригерів. Розробити логічну схему для декодування результату відліку.
8. Використовуючи JK -тригери та допоміжну логіку, розробити циклічний генератор послідовності, що задана у табл. 26.

| N | Q_2 | Q_1 | Q_0 |
|-----|-------|-------|-------|
| 1 | 0 | 0 | 1 |
| 2 | 1 | 0 | 0 |
| 3 | 0 | 1 | 0 |
| 4 | 1 | 0 | 1 |
| 5 | 1 | 1 | 0 |
| 6 | 0 | 1 | 1 |

9. На рис. 50 приведена схема цифрового пристрою з лічильниками ІЕ17. Проаналізувати і пояснити роботу пристрою.

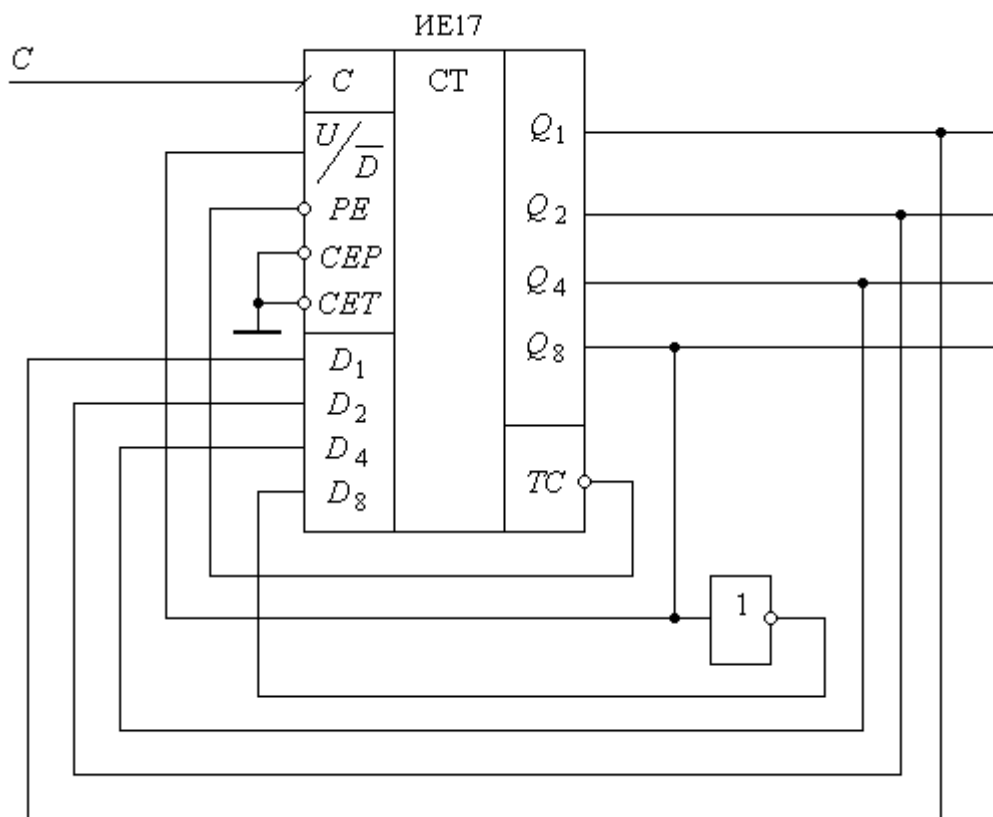


Рис. 50

10. Лічильник ІЕ18 має зовнішні з'єднання відповідно до схеми, приведеної на рис. 51. Проаналізувати особливість роботи пристрою, якщо у початковому стані маємо $Q_8 Q_4 Q_2 Q_1 = 0000$. Привести часові діаграми для одного циклу роботи лічильника.

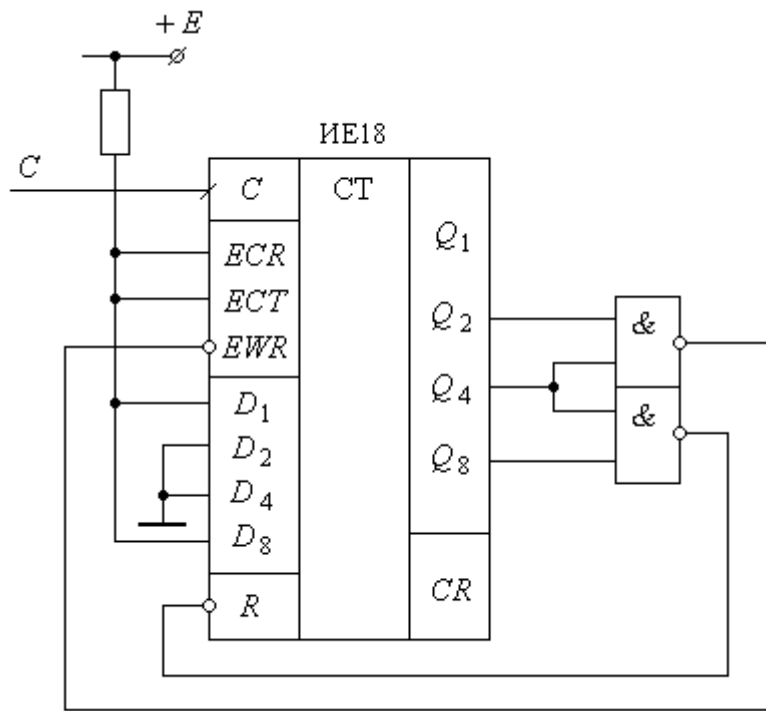


Рис. 51

11. На рис. 52 приведена схема синхронного двійкового лічильника з логікою послідовного зсуву. Привести формулу для обчислення максимальної частоти синхросигналів. У формулі повинні бути присутніми час затримки T -тригера від T -входу до Q -виходу, а також час між моментом подачі сигналу E та моментом початку фронту синхросигналу. Як називаються вказані інтервали часу?

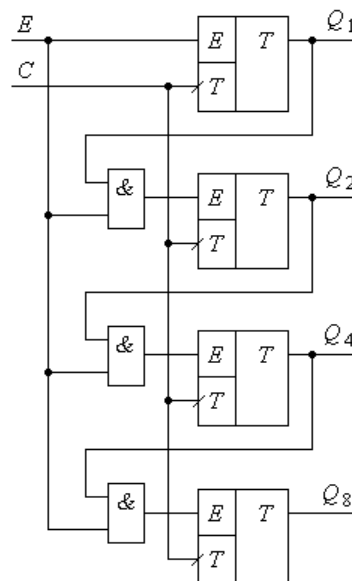


Рис. 52

12. Повторити завдання попередньої задачі для синхронного паралельного двійкового лічильника і порівняйте результати з попередніми.
13. Повторити вправу для послідовного лічильника з модулем M .
14. Використовуючи лічильник ІЕ18, спроектуйте лічильник із $M = 11$ з відліковою послідовністю: 3, 4, 5, ..., 12, 13, 3, 4, ... для роботи в циклічному режимі.
15. Використовуючи мікросхему ІЕ18, розробити схему лічильника з $M = 129$. Пояснити її роботу.
16. Використовуючи відомі мікросхеми, розробити схему лічильника з бінарним чотирьохрозрядним керуючим входом $N_3 N_2 N_1 N_0$ і одним виходом Y . Лічильник повинен видавати в циклі 16 вхідних синхроімпульсів таку кількість вихідних, яка дорівнює десятковому значенню вхідного двійкового коду.
17. Використовуючи мікросхему ІЕ17 і, якщо необхідно, допоміжну логіку, спроектувати лічильник з $M = 16$ із наступною відліковою послідовністю: 7, 6, 5, 4, 3, 2, 1, 0, 8, 9, 10, 11, 12, 13, 14, 15, 7, 6, ...
18. Розробити 8-бітний самокоригований кільцевий лічильник, в якому циркулює лише один нуль і сім одиниць в ряд.
19. Використовуючи типовий двійковий лічильник з $M = 16$ і логічні елементи “ВИКЛ. АБО”, розробити схему лічильника, вихідні сигнали якого змінюватимуться відповідно до коду Грея.
20. Розробити два варіанти лічильників з $M = 4$ на мікросхемі КР1533ТМ2 (2 синхронні D -тригери), не використовуючи інших елементів.
21. Побудувати часові діаграми у контрольних точках (C , Q_1 , Q_2) схеми лічильника з $M = 3$ (рис. 53).

мінімальний інтервал часу між двома вхідними імпульсами. Величина $f_{\max} = t_{\text{сд}}^{-1}$ називається *максимальною частотою роботи лічильника*.

Оскільки скінченні автомати можуть бути синхронними і асинхронними, то і, відповідно, ті групи автоматів, які спеціально призначені для підрахунку кількості імпульсів, можуть також бути синхронними і асинхронними.

У синхронних лічильниках, синхросигнал на всі тригери подається одночасно, а зміна стану тригера відбувається лише тоді, коли на інформаційних входах будь-якого тригера підготовлені відповідні дані. В асинхронних лічильниках, які часто називають *послідовними*, вхідна послідовність імпульсів подається лише на перший тригер, а решта тригерів спрацьовує в залежності від зміни стану попереднього.

Асинхронні лічильники. Прикладом асинхронних (послідовних) лічильників є схема, що приведена на рис. 54, *а*.

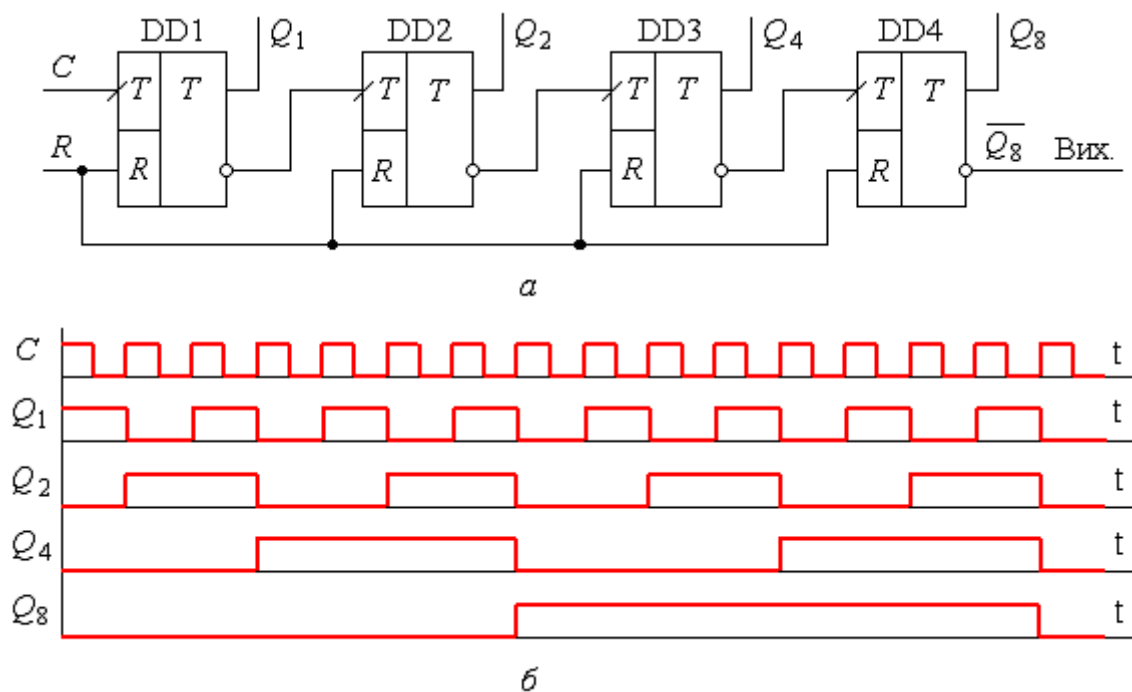


Рис. 54

Такі лічильники в своїй роботі використовують властивості *T*-тригера, оскільки вони можуть як зберігати свій стан, так і додавати за модулем 2 вхідний сигнал до інформації, записаної попередньо. Одиночний *T*-тригер ділить на 2 частоту вхідної послідовності імпульсів. Послідовне включення *n*

таких тригерів дає можливість поділити частоту входних імпульсів у $M = 2^m$ разів, або утворює лічильник з коефіцієнтом перерахунку M (модуль рахунку, ємність лічильника).

Робота найпростішого двійкового лічильника (рис. 54, *a*) пояснюється даними, приведеними у табл. 27 і часовими діаграмами, зображеними на рис. 54, *б*.

Таблиця 27

| Вхідні імпульси (N) | Q_8 | Q_4 | Q_2 | Q_1 |
|-------------------------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |
| 10 | 1 | 0 | 1 | 0 |
| 11 | 1 | 0 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 |
| 14 | 1 | 1 | 1 | 0 |
| 15 | 1 | 1 | 1 | 1 |

Перед подачею входних імпульсів тригери лічильника обнуляються. При подачі першого імпульсу на вхід C тригер, виконаний на мікросхемі DD1, у відповідності до його алгоритму роботи, змінює свій стан на протилежний. При цьому на його прямому виході з'явиться сигнал, еквівалентний логічній “1”, а на інверсному виході встановлюється логічний “0”. У відповідності до схеми, тригери з динамічною синхронізацією спрацьовують за фронтом входного імпульсу. Стани решти тригерів не зміняться, оскільки другий тригер DD2 по своєму тактовому входу сприйме перепад з “1” в “0”, що матиме місце на інверсному виході першого тригера, а тригери DD3 і DD4 на своїх входах не матимуть ніяких змін. У результаті після першого входного імпульсу значення

виходів тригерів відповідатимуть значенням, приведеним у табл. 27 в другому рядку. Другий вхідний імпульс призведе до повторної зміни стану тригера DD1. Тепер на його інверсному виході матиме місце зміна стану з логічного “0” на логічну “1”, що сприймається тригером DD2 як фронт вхідного імпульсу. В результаті тригер DD2 змінить свій стан на протилежний, і на його виході Q_2 з’явиться сигнал логічної “1”, у той час як вихід Q_1 тригера DD1 перейде в нульовий стан (в умовних позначеннях виходів лічильників прийнято номери виходів поєднувати з їх ваговими коефіцієнтами). Цей стан відповідатиме третьому рядку таблиці, відповідно до якого виходи тригерів зафіксують число 2 у двійковому коді. Третій імпульс знову змінить лише стан тригера DD1, прямий вихід якого відповідає молодшому розряду двійкового коду, що відображає кількість імпульсів, що були подані на вхід С-.

Якщо на вхід лічильника подано N імпульсів, то їх кількість, підрахована за допомогою лічильника, відповідає формулі:

$$N = K M + a_0 2^0 + a_1 2^1 + \dots + a_{n-1} 2^{n-1} = n M + \sum_{i=0}^{n-1} a_i 2^i, \quad \text{де } K - \text{кількість}$$

вихідних імпульсів лічильника; $a_i \in 1, 0$ – рівень сигналу на i -ому виході лічильника; 2^i – вагові коефіцієнти кожного прямого виходу. Для урахування вагових коефіцієнтів виходи лічильника нумерують відповідними індексами. Це дає можливість з послідовності значень виходів $Q_8 Q_4 Q_2 Q_1$, рівних, наприклад, **1011**, одразу ж зчитувати вміст лічильника, тобто кількість імпульсів, яка менша M .

При $N = M$ усі тригери лічильника обнуляються, і такий перехід фіксується зміною стану інверсного виходу тригера DD4 з “0” в “1”, що наступними аналогічними схемами повинно сприйматись як фронт вхідного імпульсу. Це дає можливість безпосереднього нарощування розрядності лічильників шляхом прямого з’єднання входу наступного лічильника з виходом попереднього. При однотипних лічильниках кількість тригерів подвоюється, і загальний коефіцієнт перерахунку визначатиметься формулою: $M_3 = 2^{2^m} = 2^m \cdot 2^m = M \cdot M$.

Звідси витікає, що при безпосередньому нарощуванні кількості лічильників загальний коефіцієнт перерахунку визначається добутком відповідних коефіцієнтів окремих лічильників.

Зазвичай код, що визначається вихідними станами тригерів, змінюється у зростаючому напрямку (*додаючи лічильники*). Операція збільшення вмісту лічильника на одиницю називається *інкрементуванням*. При зворотній зміні станів лічильники звуться *віднімаючими*, а якщо напрямок рахунку може змінюватися, то лічильник має назву *реверсивного*. Операція зменшення вмісту лічильника на одиницю називається *декрементуванням*. Напрямок рахунку визначається як динамічними властивостями тригерів (перемикання за фронтом чи за спадом), так і способом з'єднання виходів попереднього тригера зі входом наступного. Якщо, наприклад, у лічильнику використовуються тригери, які змінюють свій стан за фронтом синхроімпульсу, то для організації додаючого лічильника необхідно з'єднати вхід наступного тригера з інверсним виходом попереднього, а для створення віднімаючого лічильника для зв'язку використовуються прямі виходи тригерів. При застосуванні тригерів, що змінюють свій стан за спадом синхроімпульсу, вказані зв'язки повинні бути протилежними.

Якщо всі тригери лічильника встановлені в одиничний стан, то при подачі першого вхідного імпульсу тригер DD1 змінить свій стан на протилежний, і на виході Q_1 запишеться значення $Q_1 = 0$. Тригер DD2 при цьому не змінить свій стан, оскільки на його вході матиме місце спад імпульсу, на який він не реагує. Наступний вхідний імпульс призведе до появи на Q_1 сигналу одиничного рівня, при цьому тригер DD2 змінить свій стан, і на виходах тригерів зафіксується число $Q_8 Q_4 Q_2 Q_1 = 1101_2 = 13_{10}$. Подібний процес продовжуватиметься до повного обнуління тригерів, а шістнадцятий імпульс забезпечить встановлення всіх тригерів в одиничний стан.

Для чотирьохрозрядного додаючого лічильника з $M = 2^4 = 16$ початковим кодом, записаним на прямих виходах тригерів, є **0000**, а кінцевим, після якого

настає переповнення, – **1111**. Останній код, звичайно, є початковим для віднімаючого лічильника. У випадку, коли в якості початкового коду віднімаючого лічильника прийнятий стан **0000**, то поточні стани тригерів лічильника відображають від’ємне число зчитаних імпульсів, що представлене в допоміжному коді.

На рис. 55 приведена схема асинхронного реверсивного лічильника.

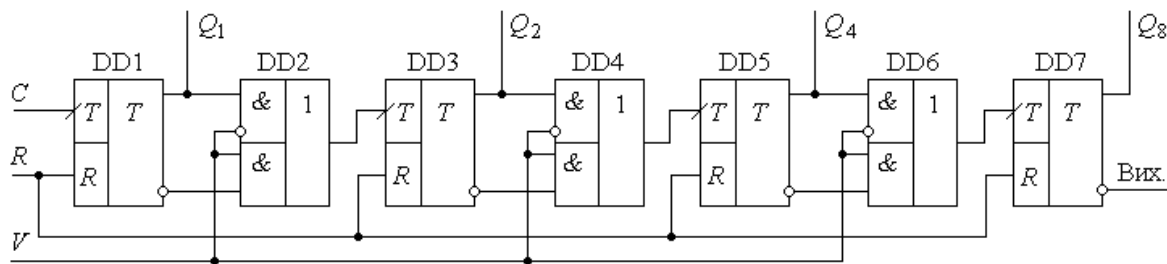


Рис. 55

Керуючий вхід V за допомогою двохвходових мультіплексорів DD2, DD4, DD6 забезпечує комутацію виходів тригерів DD1, DD3, DD5 на входи наступних – відповідно, DD3, DD5, DD7. При значенні сигналу на цьому вході $V = 1$ забезпечуватиметься підключення інверсних виходів і лічильник працюватиме на інкрементування вмісту, при $V = 0$ забезпечуватиметься режим декрементування.

R – асинхронний статичний вхід загального скидання, за яким прямі виходи всіх тригерів встановлюються в “0”.

За один цикл роботи лічильник приймає 2^m станів. Тому, з точки зору теорії скінченних автоматів, розглянутий лічильник – це асинхронний скінченний автомат з замкнутим циклом роботи, в якому перехід з одного стану до іншого забезпечується вхідними інформаційними імпульсами.

Особливість розглянутих схем лічильників полягає в тому, що для зміни значення одного з старших розрядів кодового слова повинні змінюватись і всі молодші розряди. Фактично це означає, що перед зміною стану тригера, що зберігає інформацію про вміст старших розрядів кодового слова, повинні змінити свій стан і всі попередні тригери. Якщо прийняти, що зміна стану одного тригера проходить за інтервал часу затримки t_3 , то між моментом подачі

вхідного імпульсу і зміною стану тригера старшого розряду має місце часова затримка $t_{3M} = m t_3$. Іноді цей параметр називається *часовою спроможністю лічильника*. Фізично параметр t_{3M} визначає мінімальний інтервал часу між двома вхідними імпульсами, при якому лічильник в найгіршому режимі перемикання встигає їх відпрацювати. Ця максимально можлива величина часової затримки визначає швидкодію лічильників при перерахунку імпульсних послідовностей. Максимально можлива частота вхідної послідовності імпульсів визначається за формулою: $f_M = \frac{1}{m t_3}$.

Тому послідовний спосіб забезпечення перерахунку суттєво обмежує швидкодію двійкових лічильників. Пристрої такого типу називаються *лічильниками з послідовним переносом*.

Існують лічильники з іншими видами кодування станів виходів тригерів – наприклад, *унітарним* – коли стан лічильника представляється кількістю записаних в ньому одиниць; *одинарним* – коли стан лічильника визначається розташуванням однієї одиниці.

Лічильники з довільним модулем рахунку. Лічильники з довільним модулем рахунку мають значення M , що відрізняється від цілого ступеню числа 2. Прикладами таких лічильників можуть служити пристрої з $M = 10$; $M = 12$; $M = 24$; $M = 60$ і т. д. На практиці доводиться мати справу з лічильниками, призначеними для ділення частоти вхідних послідовностей імпульсів у сотні, тисячі і десятки тисяч разів, і далеко не завжди коефіцієнт ділення може бути кратним 2^m (m – ціле число).

При побудові лічильників цього типу використовують такі способи:

- виключення зайвих станів;
- зворотного зв'язку;
- кратних модулів.

Найбільшого розповсюдження набув спосіб виключення зайвих станів. Його реалізація переважно здійснюється наступним чином:

- застосуванням попередньої установки лічильника;
- використанням примусового його обнуління.

На рис. 56 приведена схема лічильника, в якому попереднє завантаження початкового стану забезпечується через асинхронні S -входи T -тригерів за допомогою логіки DD1...DD4 і керуючого входу PE (паралельного завантаження). Через входи D_1, D_2, D_4, D_8 у лічильник може бути записаний будь-який двійковий код у діапазоні **0000 – 1111**, значення якого буде зафіксоване на виходах Q_1, Q_2, Q_4, Q_8 . Запис коду забезпечується до початку подачі вхідної послідовності імпульсів на C -вхід. Тому з моменту подачі вхідних імпульсів лічильник рахуватиме, починаючи не з нуля, а з занесеного коду.

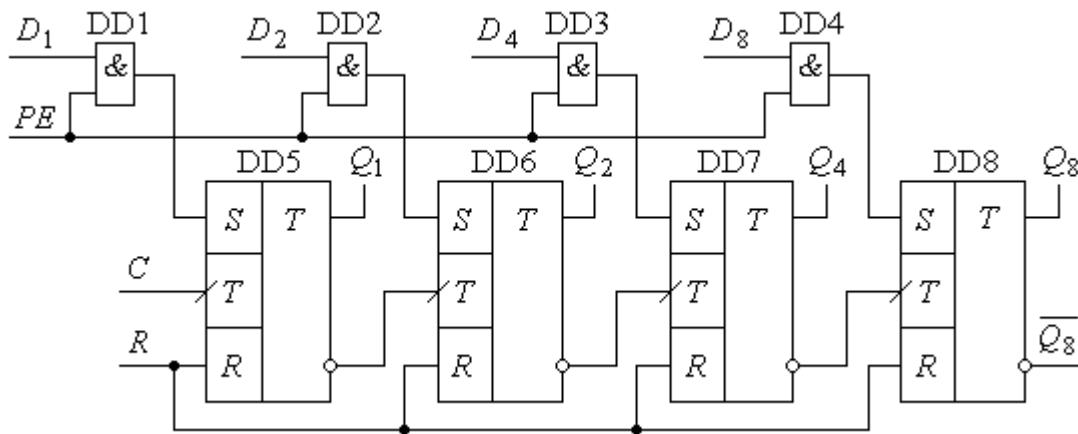


Рис. 56

Кількість імпульсів N , що може бути подана на C -вхід лічильника до переповнення, обчислюється за формулою: $N = M - D$, тобто з M станів лічильника виключається D перших станів.

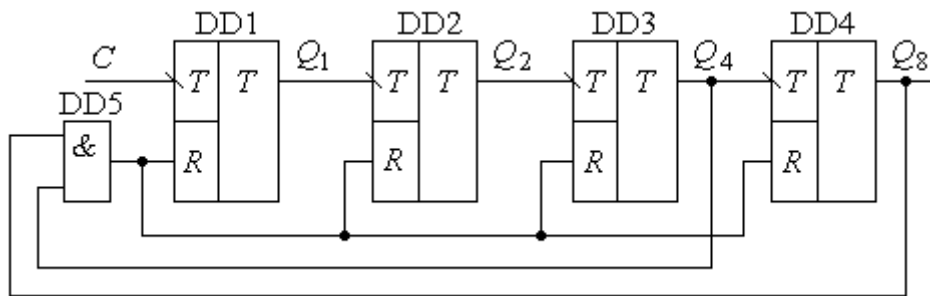
Примусове обнуління характеризується тим, що із загальної кількості станів M виключаються ті, які своїми значеннями перевищують встановлений модуль перерахунку M_B . Якщо, наприклад, лічильник з $M = 16$ повинен мати $M_B = 12$, то всі значення, котрі перевищують 12, повинні бути виключені. Тобто створення лічильника з модулем перерахунку M_B досягається виконанням двох умов:

- обнуління лічильника при досягненні поточним кодом значення M_B ;

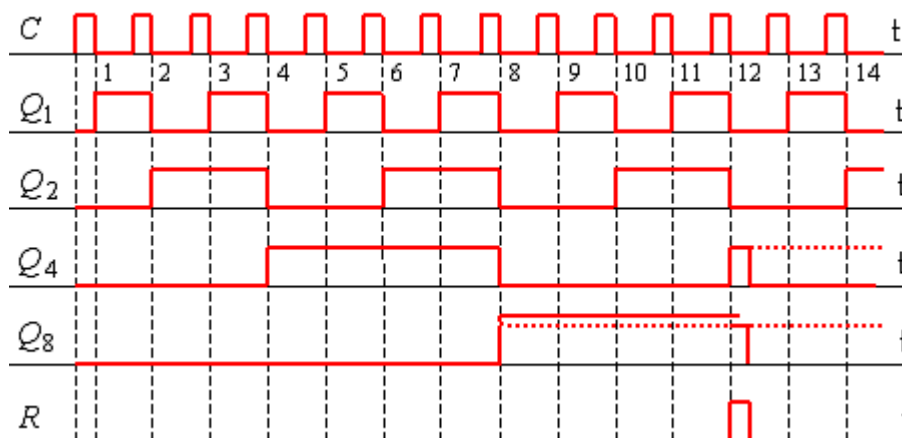
- виключення всіх станів, що перевищують M_B .

Перша умова забезпечується, якщо сигнал обнуління: $R = M_B$.

Для лічильника з $M_B = 12$ маємо: $R = Q_8 Q_4 \overline{Q_2} \overline{Q_1}$.



a



б

| | | | | | |
|-----------|----|-----------|----|----|----|
| | | $Q_2 Q_1$ | | | |
| $Q_8 Q_4$ | 00 | 00 | 01 | 11 | 10 |
| | 00 | 0 | 1 | 3 | 2 |
| | 01 | 4 | 5 | 7 | 6 |
| | 11 | 12 | 13 | 15 | 14 |
| | 10 | 8 | 9 | 11 | 10 |

в

Рис. 57

На рис. 57, *a* приводиться приклад схеми двійкового лічильника з $M_B = 12$, а на рис. 57, *б* – часові діаграми, що пояснюють його роботу. Як витікає з часових діаграм, за зрізом дванадцятого входного імпульсу на виходах Q_4 та Q_8 з'являються одиничні сигнали, які створюють сигнал обнуління R усіх тригерів.

Враховуючи те, що стан лічильника $Q_1, Q_2, Q_3, Q_4 = 0011$ є короточасним (він існує протягом часу обнуління тригерів лічильника). Його можна вважати перехідним і віднести до неробочих.

Таким чином функція виключення зайвих станів, яка забезпечує обнуління лічильника, виглядає наступним чином: $Y_B = \vee 12, 13, 14, 15$, яка після мінімізації за допомогою карти Карно (рис. 57, в) прийме вигляд $Y_B = R = Q_8 Q_4$.

Спосіб зворотного зв'язку використовується досить рідко. Особливість цього способу полягає у наступному. Припустимо, що лічильник містить у собі n тригерів. Виділимо з цієї групи меншу – з n_2 тригерами, тоді $n = n_1 + n_2$. У виділеній групі заводиться зворотній зв'язок так, щоб при появі одиниць на виходах тригерів перший тригер скидався в нуль. Тоді коефіцієнт перерахунку в цій групі становитиме $2^{n_2} - 1$, а загальний коефіцієнт перерахунку: $M = 2^{n_1} \cdot (2^{n_2} - 1) = 2^n - 2^{n_2}$.

Спосіб кратних модулів пояснюється схемою, що приведена на рис. 58.

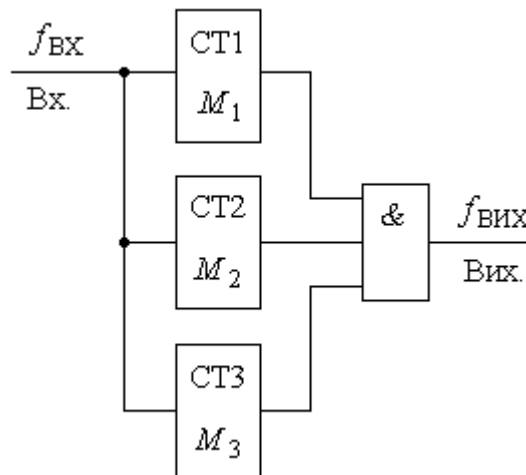


Рис. 58

Лічильники СТ1, СТ2, СТ3 з модулями рахунку, відповідно, M_1, M_2, M_3 з'єднані по входу паралельно, а їх виходи об'єднані елементом **ЗІ**. Імпульс на виході з'явиться лише тоді, коли одночасно на виходах всіх лічильників з'являться вихідні імпульси. У такій схемі частота імпульсів вхідної

послідовності $f_{\text{вх}}$ буде ділитися на коефіцієнт $M = M_1 M_2 M_3$, і частота вихідних імпульсів матиме значення: $f_{\text{вих}} = \frac{f_{\text{вх}}}{M}$.

Синхронні лічильники. В якості скінченних автоматів розглядаються лічильники, в яких синхросигнали та інформаційні сигнали, що записуються в тригери, розділені по входу. В таких схемах інформаційний сигнал, що записується у тригер, підготовляється одразу після встановлення попереднього значення, а синхросигнал подається паралельно на всі тригери. Час підготовки $t_{\text{п}}$ – це час, який необхідно враховувати як час розповсюдження сигналу через допоміжні логічні елементи, і він завжди має величину, значно меншу, ніж час перемикавання одиночного тригера. Тому швидкодія подібного типу лічильників має бути значно більшою, ніж швидкодія лічильників з послідовним переносом. Лічильники такого типу називаються *лічильниками з паралельним переносом*.

Виходячи з цього, розглянемо синтез синхронного лічильника з паралельним переносом з використанням *JK*-тригерів, який забезпечить коефіцієнт $M = 8$.

Таблиця станів лічильника, а також сигнали управління J та K входів тригерів приведені у табл. 28.

Таблиця 28

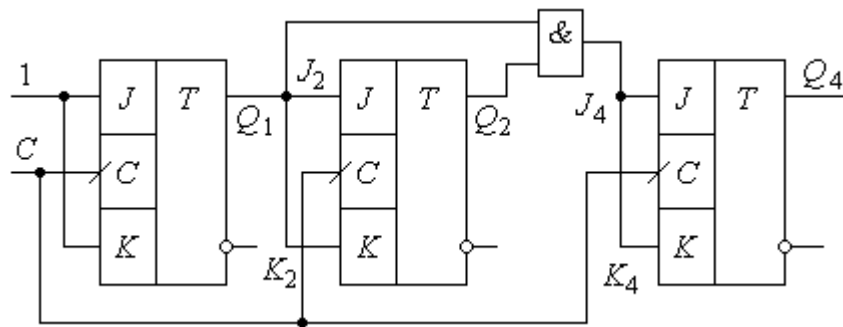
| C_n | Q_{4n} | Q_{2n} | Q_{1n} | $Q_{4(n+1)}$ | $Q_{2(n+1)}$ | $Q_{1(n+1)}$ | J_4 | K_4 | J_2 | K_2 | J_1 | K_1 |
|-------|----------|----------|----------|--------------|--------------|--------------|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | * | 0 | * | 1 | * |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | * | 1 | * | * | 1 |
| 2 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | * | * | 0 | 1 | * |
| 3 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | * | * | 1 | * | 1 |
| 4 | 1 | 0 | 0 | 1 | 0 | 1 | * | 0 | 0 | * | 1 | * |
| 5 | 1 | 0 | 1 | 1 | 1 | 0 | * | 0 | 1 | * | * | 1 |
| 6 | 1 | 1 | 0 | 1 | 1 | 1 | * | 0 | * | 0 | 1 | * |
| 7 | 1 | 1 | 1 | 0 | 0 | 0 | * | 1 | * | 1 | * | 1 |

| | | | | | | | | | | | | | | |
|-----------------|----|----|----|----|-------------|----|----|----|----|-----------|----|----|----|----|
| J_4 | | | | | J_2 | | | | | J_1 | | | | |
| $Q_2 Q_1$ | 00 | 01 | 11 | 10 | $Q_2 Q_1$ | 00 | 01 | 11 | 10 | $Q_2 Q_1$ | 00 | 01 | 11 | 10 |
| Q_4 | 0 | 0 | 1 | 0 | Q_4 | 0 | 0 | 1 | * | Q_4 | 0 | 1 | * | 1 |
| | 0 | 0 | 1 | 0 | | 0 | 0 | 1 | * | | 0 | 1 | * | 1 |
| | 1 | * | * | * | | 1 | 0 | * | * | | 1 | 1 | * | * |
| | | | | | | | | | | | | | | |
| $J_4 = Q_2 Q_1$ | | | | | $J_2 = Q_1$ | | | | | $J_1 = 1$ | | | | |

| | | | | | | | | | | | | | | |
|-----------------|----|----|----|----|-------------|----|----|----|----|-----------|----|----|----|----|
| K_4 | | | | | K_2 | | | | | K_1 | | | | |
| $Q_2 Q_1$ | 00 | 01 | 11 | 10 | $Q_2 Q_1$ | 00 | 01 | 11 | 10 | $Q_2 Q_1$ | 00 | 01 | 11 | 10 |
| Q_4 | 0 | * | * | * | Q_4 | 0 | * | * | 1 | Q_4 | 0 | * | 1 | * |
| | 0 | * | * | * | | 0 | * | * | 1 | | 0 | 1 | 1 | * |
| | 1 | 0 | 0 | 1 | | 1 | * | 1 | 1 | | 1 | * | 1 | 1 |
| | | | | | | | | | | | | | | |
| $K_4 = Q_2 Q_1$ | | | | | $K_2 = Q_1$ | | | | | $K_1 = 1$ | | | | |

Рис. 59

Використовуючи карти Карно для входів $J_4, K_4, J_2, K_2, J_1, K_1$ (рис. 59), знаходимо вирази для сигналів їх збудження. Відповідно, схема лічильника приведена на рис. 60.



Узагальнюючи отриманий результат, можна стверджувати, що, якщо для лічильника з $M = 2$:

$$J_1 = 1; K_1 = 1;$$

для лічильника з $M = 4$:

$$J_2 = Q_1; K_2 = Q_1;$$

для лічильника з $M = 8$:

$$J_4 = Q_1 Q_2 = J_2 Q_2; K_4 = Q_1 Q_2 = K_2 Q_2;$$

то для лічильника з $M = 16$ матимемо:

$$J_8 = Q_1 Q_2 Q_4 = J_4 Q_4; K_8 = Q_1 Q_2 Q_4 = K_4 Q_4.$$

Відповідно, для лічильника з $M = N$:

$$J_N = Q_1 Q_2 \dots Q_{(N-1)} = J_{(N-1)} Q_{(N-1)}; \quad K_N = Q_1 Q_2 \dots Q_{(N-1)} = K_{(N-1)} Q_{(N-1)}.$$

Лічильник виготовлений на D -тригерах DD4...DD7, які працюють у режимі T -тригерів. Допоміжна логіка DD1...DD3 забезпечує швидкий доступ послідовності імпульсів, що підраховуються, до синхровходу. При подачі сигналу високого рівня на R -вхід усі прямі виходи тригерів Q_1, Q_2, Q_4, Q_8 встановлюються у нульовий стан. За зрізом першого тактового сигналу вихід Q_1 тригера DD4 встановлюється в “1” і цим підготовлює доступ синхросигналу до синхровходу тригера DD5. Другий імпульс своїм зрізом встановить тригер DD4 у початковий стан, а DD5 – в “1”.

Цей процес триватиме доти, поки всі тригери не будуть встановлені у стан, при якому $Q_1 = Q_2 = Q_4 = Q_8 = 1$. Наступний за цим (шістнадцятий) синхроімпульс встановить всі тригери в нуль, і цикл закінчиться.

Як впливає з роботи лічильника, всі тригери сприймають імпульси вхідної послідовності практично одночасно, аналогічно відбувається зміна їх станів, що забезпечує високу швидкодію лічильника. Фактично вона визначається часом перемикавання тригера і незначною, порівняно з ним, затримкою вхідної логіки.

З аналізу роботи лічильника бачимо, що перемикавання кожного наступного тригера при приході чергового імпульсу має місце лише тоді, коли всі попередні тригери встановлені в “1”. Математично це можна виразити наступною формулою: $Q_{i(n+1)} = \overline{Q_{in}} p_i + Q_{in} \overline{p_i} = Q_{in} \oplus p_i$, де Q_{in} – стан i -го тригера до подачі імпульсу; $p_i = Q_{0n} \cdot Q_{1n} \cdot \dots \cdot Q_{(i-1)n}$ – узагальнений сигнал перенесення.

Недоліки лічильників з паралельним перенесенням полягають, перш за все, у критичності до тривалості вхідних імпульсів та їх фронтів. Тому при використанні таких пристроїв слід звертати увагу на довідкові дані по цим параметрам. Обумовлено це тим, що, незважаючи на те, що тригери лічильника перемикаються майже одночасно і час установки двійкового коду $t_{ку}$ на виході лічильника фактично визначається часом установки тригера t_T , але для підготовки до наступного перемикавання необхідно, щоб пройшов деякий час,

який називається часом підготовки t_{Π} . За цей час послідовно формуються сигнали перенесення на всіх входах елементів І. Величина t_{Π} залежить від встановленого в попередньому такті коду.

Складність практичної реалізації лічильника з великим значенням модуля M є другим їх недоліком. Це обумовлено ускладненням логіки переносу з великою кількістю входів.

Задача нарощування лічильників розв'язується декількома шляхами. Наприклад, у мікросхемі К1561ІЕ10 синхровхід запаралелюється через елемент **АБО** допоміжним інверсним входом \overline{ES} . Цей вхід дає можливість організувати рахунок як за фронтом імпульсу, так і за зрізом, а при послідовному нарощуванні використовується як вхід асинхронного вводу від виходу Q_8 попереднього лічильника. Такий спосіб нарощування називається *комбінованим (послідовно-паралельним)*.

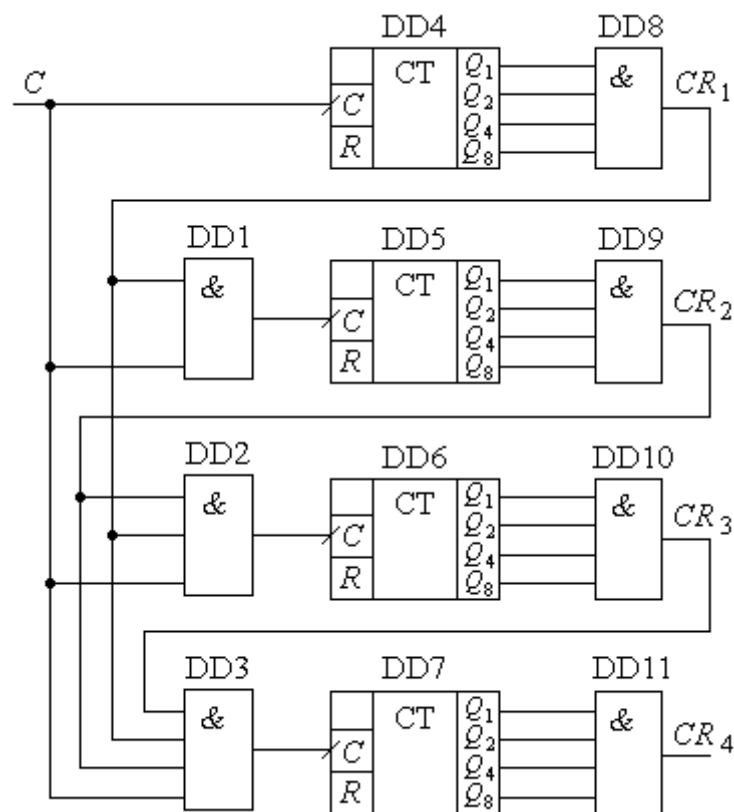


Рис. 62

На практиці часто використовуються структури з *комбінованим паралельно-паралельним перенесенням*. Особливість такого перенесення

ілюструється рис. 62. За аналогією зі схемою рис. 61, логічні елементи DD1...DD3 забезпечують паралельне перенесення від лічильників DD4...DD7. Формування сигналів перенесення $CR_1...CR_4$ з кожного лічильника виконують логічні елементи DD8...DD11 при умові, що виходи $Q_1...Q_8$ кожного з лічильників встановлені в "1". У такій структурі лічильника швидкодія визначається часом установки коду $t_{КУ}$. Ця величина практично визначається часом установки коду в межах одного лічильника. Час підготовки $t_{П\max}$ у лічильнику визначається сумою затримок у логічних елементах I. Тому максимальна частота перемикання визначатиметься як:
$$f_{\max} = \frac{1}{t_{КУ} + t_{П\max}}.$$

Серійні мікросхеми лічильників та їх використання. Властивості лічильників в основному визначаються тими зв'язками, які закладені в них для передачі вхідної послідовності імпульсів на входи тригерів, а також для передачі сигналів стану тригерів молодших розрядів на інформаційні входи наступних. Найчастіше використовують такі типи зв'язків: *безпосередній*; *за допомогою кіл послідовного перенесення*; *за допомогою кіл паралельного перенесення*.

Методологія побудови лічильників ТТЛ- і КМОН-технологій майже однакова, але більші можливості інтеграції КМОН-технологій дозволяють розширити функціональні можливості цієї елементної бази.

Лічильники ТТЛ. Прикладом пристрою з безпосереднім зв'язком є лічильник КР1533ІЕ5, схема якого приведена на рис. 63.

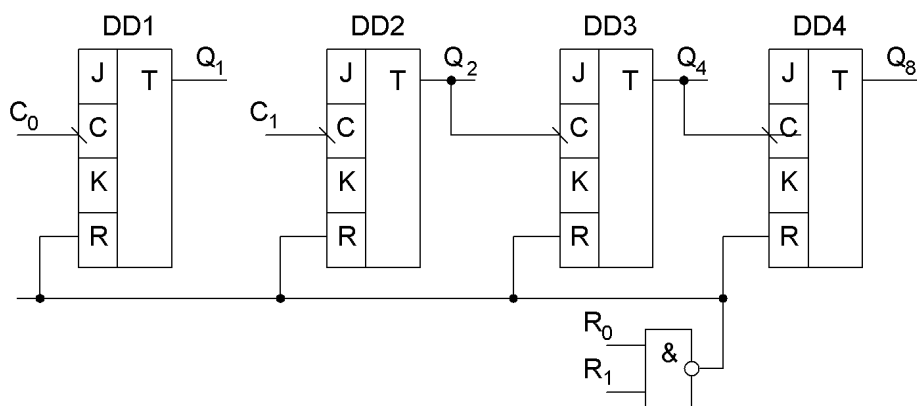


Рис. 63

Лічильник складається з чотирьох JK -тригерів DD1...DD4, які працюють в режимі T -тригерів. Тригер DD1 має вільний вихід і може використовуватись для ділення на 2 частоти імпульсів, що поступають на вхід C_0 . Тригери DD1...DD4 утворюють лічильник з коефіцієнтом перерахунку $M=8$ для імпульсної послідовності, що подається на вхід C_1 . При з'єднанні виходу Q_1 тригера DD1 з входом C_1 тригера DD2 утворюється лічильник з $M=16$. Входи R_0 і R_1 забезпечують два можливі режими: *режим блокування* та *режим перерахунку*.

Нарощування таких лічильників з метою збільшення коефіцієнта M досягається шляхом з'єднання виходу Q_8 одного лічильника з входом C_0 або C_1 іншого.

Для обнуління станів лічильника використовуються входи R_0 і R_1 , які також є асинхронними, і при високому рівні сигналу на них всі виходи лічильника приймають нульовий (низький) рівень.

Використовуючи об'єднання по I входів обнуління R_0 і R_1 , лічильник забезпечує можливість створити будь-який модуль перерахунку (за винятком $M=3$) від двох до шістнадцяти.

Аналогічну структуру мають лічильники ІЕ2 та ІЕ4. Різниця полягає в тому, що в ІЕ2 три тригери об'єднані в схему, що дає $M=5$, а в ІЕ4 – аналогічно, три тригери забезпечують модуль $M=6$. Перша мікросхема легко реалізує схему ділення на 10, а друга – схему ділення на 12.

Близькі за принципом роботи двійкові асинхронні лічильники – мікросхеми ІЕ19. В них розміщується два лічильники з динамічним входом (відлік за зрізом) і потенційним низькорівневим асинхронним входом обнуління.

Лічильник ІЕ14 має структуру, аналогічну лічильнику ІЕ2 (тригери розділені на дві групи: на одному тригері забезпечується коефіцієнт ділення на 2, а на решті трьох – коефіцієнт ділення на 5). У той же час, ІЕ14 має значно

більші функціональні можливості завдяки використанню *JK*-тригерів та допоміжної керуючої логіки.

Лічильник ІЕ15 – двійковий лічильник з попередньою установкою, асинхронний, виконаний за структурою, подібною до ІЕ14.

Лічильники, які виготовляються за схемами, подібними до ІЕ5, характеризуються простотою, легкістю нарощування, високою надійністю при змінній частоті і тривалості входних імпульсів. Їх недолік полягає у значній затримці перемикавання, величина якої залежить від коефіцієнта перерахунку.

Прикладом синхронних лічильників з попередньою установкою є мікросхеми ІЕ6 та ІЕ7. Обидва вони реверсивні, ІЕ6 – двійково-десятковий, а ІЕ7 – двійковий.

Своєрідним лічильником є мікросхема ІЕ8, яка має спеціальне призначення – програмоване ділення частоти входних імпульсних послідовностей. Найбільш широко використовується кратне ділення частоти на 2^i , оскільки в такому випадку отримується симетрична вихідна послідовність. Але мікросхема забезпечує довільний (у заданих межах) коефіцієнт ділення, при цьому вихідна послідовність матиме різні інтервали часу між імпульсами.

Для обнуління лічильника використовується асинхронний вхід R , високий рівень на якому при одночасному високому рівні на вході \bar{S} забезпечує обнуління всіх тригерів лічильника, а також переведення виходу $\bar{C}_{\text{вих}}$ і Q у стан високого потенціалу.

Сигнал дозволу, що подається на вхід \bar{EI} (*Enable Input*), має низький рівень, але для забезпечення режиму ділення частоти необхідно, щоб одночасно і на вхід \bar{S} також подавався сигнал низького рівня. Зупинка роботи мікросхеми забезпечується високим рівнем сигналу на вході \bar{S} . Вихід CEP використовується для нарощування мікросхем.

Група лічильників ІЕ9, ІЕ10, ІЕ11, ІЕ13, ІЕ18 – це синхронні лічильники, близькі по своїй структурі. Лічильник ІЕ9 – двійково-десятковий, решта – двійкові.

Лічильники КМОН. Лічильники КМОН, як і ТТЛ-серій, призначені для підрахунку кількості імпульсів та ділення частоти. Лічильники для підрахунку імпульсів умовно розділяють на *спеціалізовані* – призначені для використання в електронних годинниках, таймерах, пристроях для організації часових затримок, а також *універсальні* – пристрої загального призначення.

Мікросхеми серії 176 (ІЕ1...ІЕ8) були замінені більш пізніми серіями КР561 (CD4000А) та КР1554 (74АС), хоча група мікросхем ІЕ2, ІЕ3, ІЕ4, ІЕ5 знаходить досить широке використання в радіолюбительській практиці. Лічильники останніх серій знаходять досить широке використання в цифровій схемотехніці.

Слід зазначити, що, починаючи з серії 1554, функціональне призначення мікросхеми визначається її номером і є спільним як для ТТЛ, так і для КМОН. Тому, наприклад, лічильники ІЕ6 серій КР1533 і КР1554 є повністю еквівалентними за своїми функціональними можливостями. У мікросхемах основних виробників зарубіжжя така особливість закладалась з самого початку.

Найширше використання у промислових розробках знайшла серія 561 (564), яка мала свої аналоги серед мікросхем CD4000А. У цій серії мікросхеми лічильників виготовлялись як на основі схемотехніки, описаної вище, так і на основі використання лічильників Джонсона (ІЕ8, ІЕ9, ІЕ19), які будуть розглянуті пізніше.

Мікросхема 561ІЕ10 містить два незалежні чотирьохрозрядні двійкові лічильники з паралельним переносом. Умовне позначення одного з них приведене на рис. 64, а на рис. 65 зображена структурна схема, що пояснює особливості його функціонування.

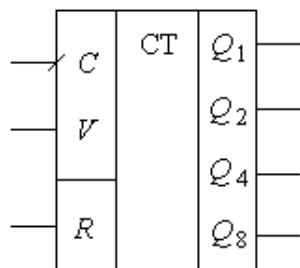


Рис. 64

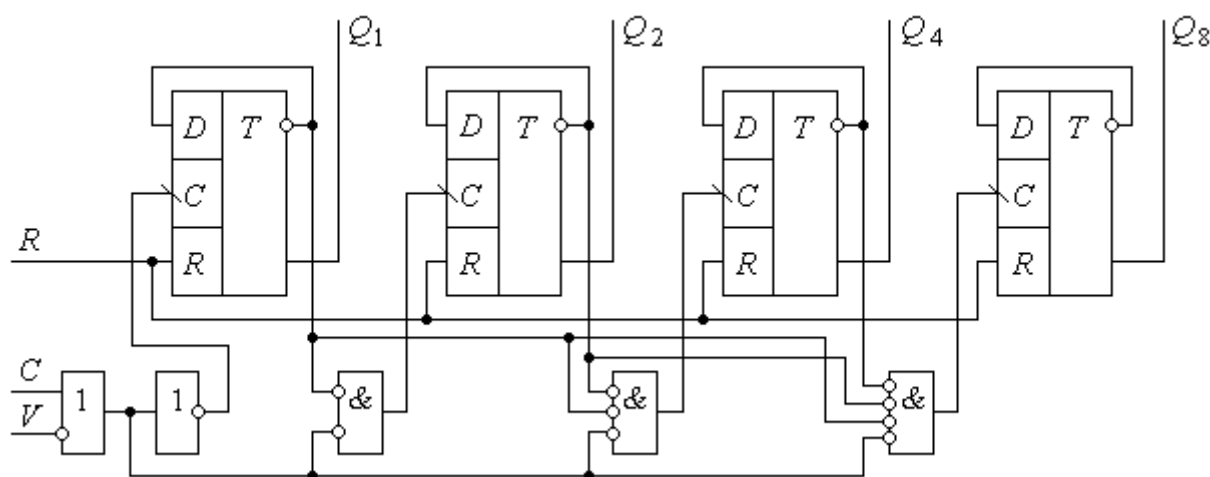


Рис. 65

Таблиця 29

| Входи | | | Режими роботи |
|-------|---|---|---------------|
| C | V | R | |
| ↑ | 1 | 0 | Режим відліку |
| 0 | ↓ | 0 | Обнуління |
| x | x | 1 | Обнуління |

Лічильник побудований на основі D -тригерів, які працюють в режимі T -тригера. Тактовий вхід C і дозволяючий V , об'єднані елементом **АБО**, взаємно інверсні. Це дає можливість взаємозамінити їх, а також організовувати відлік за фронтом чи спадом тактового імпульсу. Режими роботи лічильника приведені у табл. 29. При необхідності нарощування лічильників можна використовувати як послідовне, так і паралельне формування переносу. При послідовному вихід Q_8 попереднього лічильника необхідно з'єднати з входом V наступного, а вхід C приєднати до потенціалу низького рівня.

Лічильники ІЕ10 мають можливість зменшення модулю M . Але для забезпечення $M < 2^4$ необхідно використовувати зовнішні елементи і враховувати, що лічильник обнуляється сигналом високого рівня на вході R . Обнуління при заданому M можна забезпечити як автоматичним, так і керованим.

Мікросхеми 561ИЕ11 і 561ИЕ14 є чотирьохрозрядними реверсивними лічильниками з паралельним переносом і мають багато спільного як з мікросхемами ТТЛ (ИЕ16 і ИЕ17), так і між собою. Різниця між ними полягає лише в тому, що 561ИЕ11 має $M = 16$ і вхід R загального обнуління, а 561ИЕ14 має вхід перемикавання $M = 16 / M = 10$, але вхід R відсутній.

Мікросхема 561ИЕ156 (564ИЕ15) не має аналогів в ТТЛ ІС. Це програмований лічильник з одним виходом, який працює лише в режимі ділення. Коефіцієнт ділення K_d задається відповідними установками рівнів вхідних сигналів і може мати будь-яке значення в межах від **3** до **21327** з кроком **1**. Лічильник може працювати у двох режимах:

- *безперервної дії*, коли на виході отримується послідовність імпульсів частотою $f_{\text{вих}} = \frac{f_{\text{вх}}}{K_d}$ (з тривалістю імпульсів $t_i = \frac{1}{f_{\text{вх}}}$);
- *одноразового відліку*, при якому після подачі на вхід K_d імпульсів вихідний сигнал змінюється з низького рівня на високий.

Для визначення рівнів сигналів, що подаються на входи $J_1 \dots J_{16}$ мікросхеми, використовується формула:

$K_d = M \cdot (10^3 P_T + 10^2 P_C + 10^1 P_D + 10^0 P_O) + P_3$, де M – модуль, який може мати наступні значення: **2, 4, 5, 8, 10**; P_T – множник тисяч, приймає значення від **0** до **7**; P_C, P_D, P_O – множники сотень, десятків, одиниць; P_3 – залишок від ділення.

Основний режим роботи лічильника – режим безперервної дії, в якому по закінченню циклу у внутрішні лічильники мікросхеми перезаписуються дані з входів, встановлюючи кожен з них у відповідний коефіцієнт перерахунку.

Режим одноразового відліку характеризується тим, що його обов'язково упереджує режим установки в початкове значення. Після виконання одного циклу на виході Y встановлюється високий рівень сигналу, який зберігається незалежно від наявності вхідних імпульсів. Якщо встановити $L = 0$, то лічильник перейде в циклічний режим з частотою вихідних імпульсів $f_{\text{вих}} = \frac{f_{\text{вх}}}{N}$. Щоб

виконати повторний одиничний запуск, необхідно знову встановити лічильник у початковий стан, після чого повернутися до режиму одноразового відліку.

Контрольні питання

1. Дайте визначення терміну “лічильник”.
2. Які параметри характеризують лічильник як цифровий пристрій?
Назвіть їх і дайте фізичну інтерпретацію.
3. У чому полягає різниця між синхронними і асинхронними лічильниками?
4. Поясніть суть операцій інкрементування та декрементування вмісту лічильника.
5. Чим обмежується швидкодія лічильників з послідовним переносом?
6. Які способи використовуються для побудови лічильників з довільним модулем перерахунку?
7. Які недоліки має лічильник з попередньою установкою модуля перерахунку?
8. У чому полягає перевага синхронних лічильників перед асинхронними?
9. Перерахуйте та обґрунтуйте недоліки двійкового лічильника з паралельним переносом.
10. Якими шляхами розв’язується задача нарощування розрядності синхронних лічильників з паралельним переносом?
11. У чому полягає недолік асинхронних лічильників при їх використанні з дешифраторами та семисегментними індикаторами для відображення інформації?
12. Як на основі мікросхеми ІЕ8 розробити перетворювач чотирьохрозрядного двійкового коду в частоту вихідних імпульсів; восьмирозрядного двійкового коду в частоту вихідних імпульсів?

13. Поясніть особливості нарощування розрядності лічильників ІЕ9 та ІЕ10.

14. Чим мікросхема 564ІЕ15 принципово відрізняється від інших типів мікросхем лічильників?

15. Приведіть порівняльну характеристику переваг та недоліків двійкових лічильників і лічильників Джонсона.

16. Поясніть, на якому принципі будуються цифрові одинівбратори та різноманітні таймери.

Рекомендована література

1. Тиртишніков О.І., Корж Ю.М. Обчислювальна техніка та мікропроцесори. Частина 2. Цифрові автомати: Навчальний посібник. – Полтава: ПВІЗ, 2006, с. 54 – 62.

2. Калабеков Б.А., Мамзелев И.А. Цифровые устройства и микропроцессорные системы. М.: Радио и связь, 1987.